Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Geoscience and Petroleum

Magnus Bareksten
Lars Djuve Vambheim

Digital Well Planning

Next Generation Workflow

Master's thesis in Petroleum Engineering
Supervisor: Sigbjørn Sangesland
Co-supervisor: Bjørn Astor Brechan
June 2022

NTNU
Norwegian University of
Science and Technology

Magnus Bareksten
Lars Djuve Vambheim

Digital Well Planning

Next Generation Workflow

Master’s thesis in Petroleum Engineering
Supervisor: Sigbjørn Sangesland
Co-supervisor: Bjørn Astor Brechan
June 2022

Norwegian University of Science and Technology
Faculty of Engineering
Department of Geoscience and Petroleum

NTNU
Norwegian University of
Science and Technology

Acknowledgements

We would like to express our sincerest gratitude towards our supervisor Bjørn Astor Brechan for guiding us through the last year of our masters degrees. Your ability to be positive and enthusiastic, as well as all of the valuable input you have given us throughout the year has been greatly appreciated. Thank you for allowing us to continue the work you started on with your PhD thesis.

We would also like to thank our main supervisor Professor Sigbjørn Sangesland for guidance throughout the thesis work.

A special thanks also go to both our families for their support throughout our five years at NTNU.

Finally, we would like to thank each other for the collaboration, discussions and hard work throughout the year.

Thank you,

Magnus Bareksten & Lars Djuve Vambheim.

(Intentionally left blank.)

Summary

The petroleum industry is in a period with high focus on digitalization and automation. This focus is driven by an ambition to reduce costs, make operations more efficient and enhance safety. Several new software solutions have already been launched, with a particular focus on automating rig equipment. However, the process of planning wells is still manual, i.e., human-driven with limited automation. The research conducted in this thesis builds on principles introduced in the PhD thesis ”Framework for automated well planning and Digital Well Management” from 2020 by Bjørn Astor Brechan. This PhD outlines a model for well planning and support called Life Cycle Well Integrity Model (LCWIM). The model is a new approach to digital well planning integrating all aspects in well planning.

In the fall of 2021, a generic detailed operational procedure (DOP) for drilling a 17 1/2” section was digitalized, and the result was a digital detailed operational procedure (DDOP). The DDOP was built using the principles in the LCWIM, and it was the first step in developing a software tool for fully automated support of well planning. The research in thesis is a continuation of this work, with focus on developing a software for creating DDOPs with capacity to perform automated engineering. This software is called Well Planning Software (WPS) and the scope of the research and software was to prove various concepts in the LCWIM. A concept in the LCWIM is the event manager. It can be compared to a high-level program on a digital format that ties together all information required for well planning and operations. The DDOP can therefore be viewed as a sub-level to the event manager.

The majority of the work conducted throughout this thesis was used to develop the software and the process of learning and developing the right programming solutions. The WPS demonstrates an effective and agile way of well planning, proving various concepts from the LCWIM. It enables generic DDOPs to be adjusted to suit each individual field and integrates multiple aspects of planning, including engineering, rig automation, contracts and equipment. By allowing agile plans to be digitalized, these can be looked upon as a digital experience, which can automatically be used in subsequent planning and operations. The WPS was developed with focus on automated engineering and rig automation, to create a direct link between planning and operations. The WPS allows DDOPs to be made in a format readable for both humans and software, enabling automated engineering and automation of rig equipment in operations.

During the development of the WPS prototype, it was identified several improvements to the LCWIM concept and there are multiple proposals by the authors for future development of the software. The WPS needs to include other parts of planning, as well as support for other disciplines of the well life cycle to see more in detail and prove the impact of the LCWIM concept. Moreover, incorporating digital experiences into the WPS is viewed as a key aspect in further development. The current engineering and automation modules need improvements to be able to provide a program ready to be used in the industry. Finally, the software needs to adopt ”smart” functionality to minimize the human interaction needed to run the program.

(Intentionally left blank.)

Sammendrag

Petroleumsnæringen er inne i en periode med høyt fokus på digitalisering og automatisering. Dette fokuset er spesielt drevet av en ambisjon om å redusere kostnader, effektivisere drift og øke sikkerheten. Flere nye programvareløsninger eksisterer allerede, med et særlig fokus på å automatisere riggutstyr. Prosessen med å planlegge brønner er imidlertid fortsatt menneskedrevet med begrenset automatisering. Forskningen utført i denne oppgaven bygger på prinsipper introdusert i Ph.d.-avhandlingen "Framework for automated well planning and Digital Well Management" fra 2020 av Bjørn Astor Brechan. Denne doktorgraden skisserer en modell for brønnplanlegging og støtte kalt Life Cycle Well Integrity Model (LCWIM). Modellen er en ny tilnærming til integrering av digital brønnplanlegging, i alle aspekter av brønnplanleggingsfasen.

En genererisk detaljert operasjonsprosedyre (DOP) for boring av en 17 1/2" seksjon ble digitalisert høsten 2021, og resultatet ble en digital detaljert operasjonsprosedyre (DDOP). DDOP-en ble utviklet ved bruk av prinsippene i LCWIM, og var det første trinnet i å utvikle et programvareverktøy for helautomatisk brønnplanlegging. Forskningen i oppgaven er en viderføring av dette arbeidet, med fokus på utvikling av programvare for å lage DDOP-er med kapasitet til å utføre automatisert ingeniørarbeid. Denne programvaren kalles Well Planning Software (WPS). Omfanget av forskingen og programvaren er å bevise ulike konsepter i LCWIM. Et konsept i LCWIM er det som kalles event manager. Event manager kan sammenlignes med et høynivåprogram i digitalt format som binder sammen all informasjon som kreves for brønnplanlegging og drift. DDOP-er kan derfor sees på som et undernivå av event manager.

Den største delen av oppgaven bestod av å utvikle programvaren samt prosessen med å lære seg og utvikle de riktige programmeringsmetodene. WPS demonstrerer en effektiv og smidig måte å planlegge en brønn på, og beviser ulike konserter fra LCWIM. Den gjør det mulig å justere generiske DDOP-er for å passe hvert enkelt felt, og integrerer flere avspekter ved plannlegging inkludert ingeniørarbeid, riggautomatisering, kontrakter og utstyr. Ved å la fleksible planer digitaliseres, kan disse sees på som en digital erfaring som automatisk kan brukes i etterfølgende planlegging og drift. WPS ble utviklet med fokus på automatisert ingeniørarbeid og riggautomatisering, for å skape en direkte kobling mellom planlegging og drift. WPS gjør at konstruerte DDOP-er er i et format som er lesbart både for mennesker og for programvare. Dette muliggjør automatisert konstruksjon og automatisering av riggutsyr i operasjoner.

I utviklingen av WPS-prototypen ble det identifisert flere forbedringer av LCWIM-konseptet, og flere forslag for fremtidig utvikling av programvaren. WPS må inkludere andre deler av plannleggingsfasen, samt støtte andre disipliner i hele livssyklusen til en brønn for å se og bevise LCWIM-konseptes fulle potensial. WPS trenger også støtte for å implementere digitale erfaringer, og dette sees på som et nøkkelaspekt for videre utvikling. De nåværende ingeniør- og automasjonsmodulene trenger forbedringer for å kunne tilby et program klart til bruk i industrien. Til slutt må programvaren ta i bruk "smart" funksjonalitet for å minimere den menneskelige interaksjonen som behøves for å kjøre programmet.

(Intentionally left blank.)

List of Abbreviations

AI	Artificial Intelligence
BHA	Bottom Hole Assembly
CAPEX	Capital Expenditure
CSV	Comma-Separated Values
DDOP	Digital Detailed Operational Procedure
DOP	Detailed Operational Procedure
DWM	Digital Well Management
E&P	Exploration and Production
ECD	Equivalent Circulating Density
GUI	Graphical User Interface
HSE	Health, Safety and Environment
KPI	Key Performance Indicator
LCWIM	Life Cycle Well Integrity Model
ML	Machine Learning
MWD	Measurement While Drilling
NCS	Norwegian Continental Shelf
NoSQL	Not only Structured Query Language
NPT	Non-Productive time
OOP	Object-Oriented Programming
OPEX	Operating Expenditure
P&A	Plug and Abandonment
POOH	Pull Out Of Hole
PSA	Petroleum Safety Authority Norway

RAS	Rig Automation Software
RIH	Run In Hole
ROP	Rate Of Penetration
SQL	Structured Query Language
TVD	True Vertical Depth
WH	Well Head
WOS	Well Operative System
WPS	Well Planning Software

Nomenclature

ΔP	Pressure loss
$\overline{v}$	Mean velocity
ρ	Density
σ_{300}	Fann reading at 300 RPM
σ_{600}	Fann reading at 600 RPM
A_t	Total nozzle area
C_d	Nozzle coefficient
CCI	Hole cleaning index
d_1	Internal diameter of pipe/collar
d_2	Outer diameter of pipe/collar
d_3	Hole diameter
f	Friction factor
K	Consistency index
L	Pipe length
MW	Mud weight
n	Flow behavior index

N_{Rec}	Critical Reynold's number
q	Flow rate
v	Fluid velocity
v_m	Maximum pipe velocity
v_p	Pipe velocity
v_{ann}	Annular velocity

(Intentionally left blank.)

Table of Contents

(Intentionally left blank.)

List of Figures

List of Tables

(Intentionally left blank.)

1 Introduction

The petroleum industry has changed significantly over the last decades. Technological advancements have been made in all parts of the industry, from exploration all the way to production and refinement. Operations are now safer, more efficient and have a lower environmental footprint than ever before. The changes are a result of heavy investments. Margins have been lowered, forcing the industry to develop new and more efficient technologies and operations. Stricter regulations regarding health, safety and environment (HSE) have also been a major driving force. From being an industry where producing hydrocarbons and making money was paramount, the industry is now just as much focused on lowering emissions and enabling clean energy production.

As with almost all industries, the petroleum industry has seen a major change due to digitalization over the last decade. Work is now carried out with digital support from sophisticated applications enabling more efficient work flows. Data is shared both internally and externally in a more fluent way now, than before. In the petroleum industry specifically, digitalization has enabled remote operations and digital twins, to name two examples.

Despite the recent wave of digitalization, an area that still has a significant potential for further improvements is well planning. It is a comprehensive process, with many different departments involved. Per today, the process is still somewhat human-oriented, meaning that information is shared through written documents and meetings. Huge amounts of data are being moved back and forth between different departments and applications in the planning process. This is a cumbersome process. Making changes to the well plan at a late stage in the planning process cause a lot of extra work, as it often triggers a full recheck of all engineering.

A software facilitating more agile well planning and smoother information flow would account for significant savings in terms of cost and time spent, as well as improving the overall quality of the well plans. Minimizing the boundaries between compartments allowing for more cooperation and integrated operations will provide huge value. Moreover, experience that is utilized from similar operations is also heavily reliant on the skills and experience of people involved. Enabling previous experiences to be incorporated into new well plans with less human interaction through digital solutions will be important.

Automation is another hot topic in many areas of the industry today. This includes automation of repetitive tasks both onshore and offshore. Drilling automation is considered a key enabler for more efficient operations. Furthermore, significant value can also be created using automated solutions in well planning, reducing workload and improving quality.

This thesis outlines a prototype for well planning programmed in python. The prototype is called Well Planning Software (WPS). It is intended to showcase how different elements of well planning such as engineering, contracts and equipment can be integrated into one system. Additionally, the software shows how digital detailed operational procedures (DDOP) can be formatted to being readable for rig automation softwares (RAS) and easily read and followed by humans.

1.1 Background

This thesis is a continuation of the work carried out in the Specialization Projects during the fall semester of 2021 (Bareksten, 2021; Vambheim, 2021). These projects gave an introduction to digital well planning and how to build a digital procedure from today's analogue and paper based detailed operational procedure (DOP). The projects were based on the doctoral thesis by Bjørn Brechan (2020), which will also play a key role for the work in this thesis. The work focuses on materializing theory from the specialization project to prove various concepts presented in the PhD thesis.

The PhD outlines a model for planning and support throughout the full well life cycle. The model is called Life Cycle Well Integrity Model (LCWIM). It includes and integrates the following elements into the same model:

- Logistics
- Engineering
- Risk/method
- Management of change
- Contracts
- Program
- DOP
- Governing documentation
- Equipment
- Experience
- Other

1.2 Objective

Well planning is a large area that encompasses many disciplines. Moving from a compartmentalized structure which follows any plan on paper to a fully digital workflow opens up for a review of the entire process. Advancements must be achieved in all parts of planning. This thesis focuses on digital well planning through proving core principles of the LCWIM. However, as time is limited, this research will be restricted to development of a prototype for planning drilling operations using principles from the LCWIM. The objectives of the thesis can be summarized to:

- Prove digital well planning concepts presented in the LCWIM through development of a prototype software for well planning.
- Develop the software using object-oriented programming (OOP), to facilitate agile well planning.
- Identify a suitable approach and programming technique to develop the software to achieve the intended functionality.
- Enable the software to create digital detailed operational procedures (DDOPs) for drilling in formats readable for both humans and RAS.
- Integrate engineering, contracts and equipment directly with well plans created in the software.

1.3 Outline

Firstly, the thesis will give a theoretical introduction to important elements and concepts in the development of the well planning application.

Secondly, principles and methods applied in the development of the application will be presented. It will be explained how concepts from the previous chapter are used in this thesis specifically.

Thirdly, a walk through of the application developed will be presented.

Finally, the results obtained will be discussed, a conclusion drawn and further work proposed.

1.4 Definitions

- **Well Planning Software (WPS):** Shortening used when referring to the application developed in this thesis.
- **Digital detailed operational procedure (DDOP):** A plan describing an operation in a well, made in a digital format.
- **Life Cycle Well Integrity Model (LCWIM):** Platform for well planning and support for the well's full life cycle.
- **Digital well management (DWM):** Term used to describe the process of digital and automated well construction and integrity (Brechan, 2020).
- **Rig Automation System (RAS):** Term used when referring to software driving automated operations in drilling.
- **Digitalization:** Moving from an analogue to a digital process.
- **Automation:** In this thesis, automation is refereed to as the process of letting software perform tasks with little to no human interaction.

(Intentionally left blank.)

2 Theory

2.1 Digitalization and automation in the petroleum industry

Digitalization and automation have in later years become increasingly important in the petroleum industry. It is and will be an instrumental tool to lower cost, emissions and improve safety and effectiveness in the industry.

2.1.1 Well Planning Today

While comparable industries have modernized their workflow and optimized manufacturing efficiency and quality, the petroleum industry is still mostly text-based when it comes to the workflow in the planning phase (Brechan, 2020). One of the biggest shortcomings of the current workflow is the lack of a fully integrated software that takes all aspects in well planning into account. Figure 2.1 shows the different phases a well undergoes from it is first discovered until plug and abandonment (P&A). Going even further, the industry lacks a software in which experience from the full life cycle of wells is taken into account. Experience is not automatically transferred from the planning phase, to the construction phase, to production and finally to P&A. Moreover, experience from one well is not automatically transferred to another.

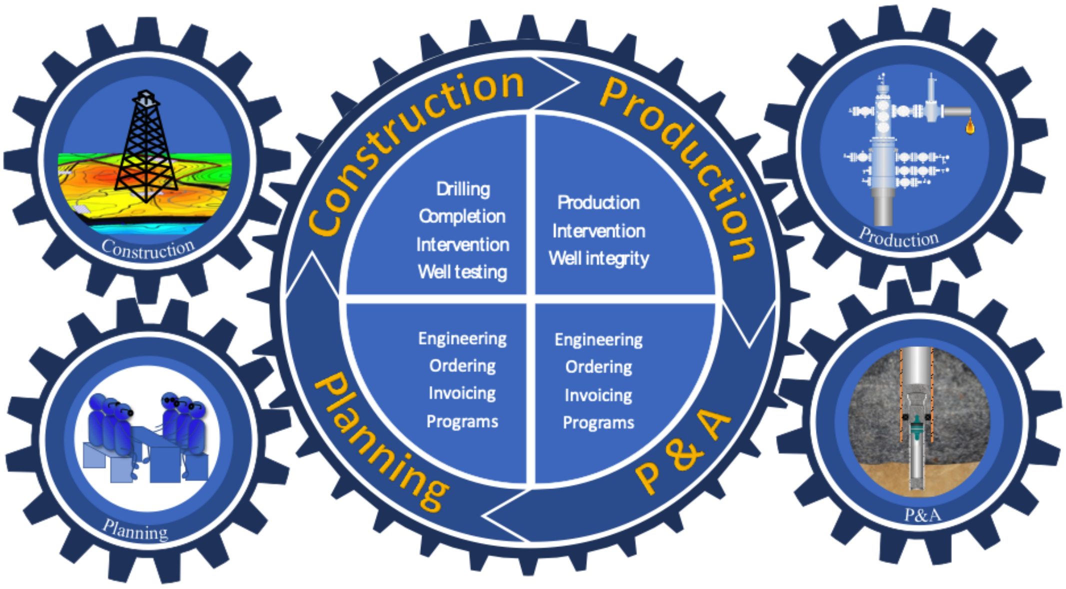


Figure 2.1: Phases during a well's life cycle (Brechan, personal communication, September 10, 2021).

In well planning today, most of an engineer's time goes to administrative work. This includes logistics, handling orders and invoices, as well as collecting and transmitting info through emails, phone calls and in meetings. Figure 2.2 shows how an engineer's time is distributed between engineering work and administrative work, with the most time consuming activity being collecting and transmitting info. This partly due to different software being used in different planning phases. Data is not automatically conveyed between software and departments. The chief data scientist for Halliburton's Landmark software division, Satyam Priyadarshy, states that one of the biggest challenges in the industry is so-called data democratization, i.e. making the data available to the right people across different departments (DiChristopher, 2015). The need for manually collecting and transmitting info might be obsolete, should a software integrating and sharing all info become available. Less time being spent on administrative work frees up time for engineers to produce better well plans.

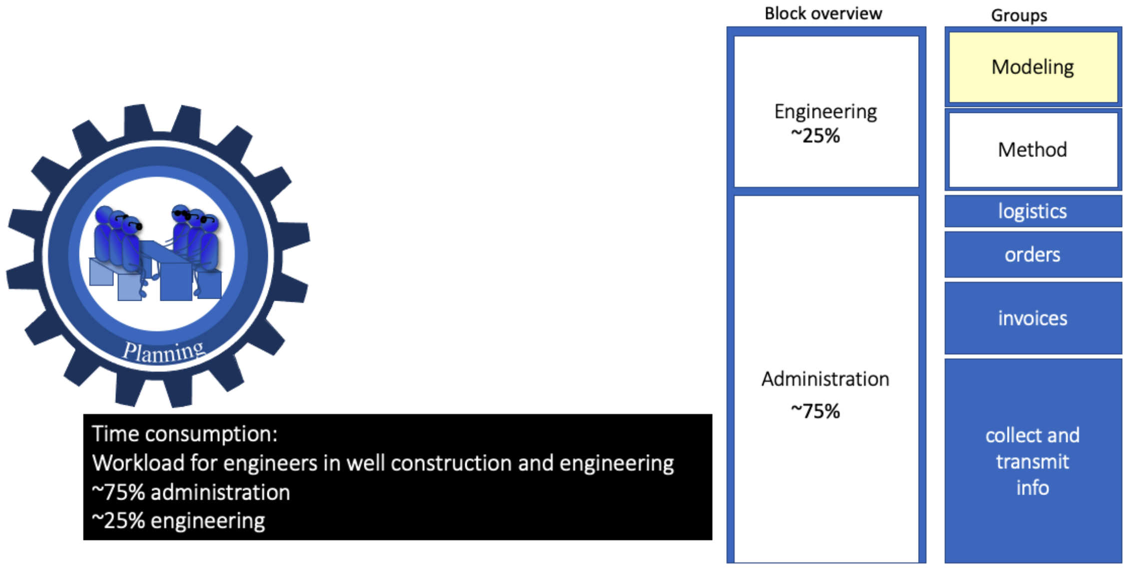


Figure 2.2: Time consumption in the planning phase (Brechan, personal communication, September 10, 2021).

Only about 25% of an engineers time is effective work addressing well planning and engineering problems, ensuring efficiency and quality in the well that is to be delivered. Different software are available for this purpose on the market today. However, none of these integrate all parts of well planning, including administrative work.

The most popular software in the industry today is Halliburton's Landmark Engineering Desktop (Landmark EDT). Landmark EDT consists of five separate software: Compass, WellPlan, CasingWear, StressCheck and WellCat. These software are responsible for different parts of the well planning process. Compass is used for determining wellpath, StressCheck, CasingWear and WellCat are used for casing and completion design, while WellPlan is used for simulating drilling, torque and drag (T&D), hydraulics and designing bottom hole assemblies (BHA). These applications do only partly communicate with each other, and a lot data must be manually inputted or moved between apps, making it an inefficient process.

Additionally, newer and more modern well planning software have entered the scene. One of these is WellDesign, an integrated cloud software for planning, created by the Norwegian company Oliasoft (Oliasoft, 2022). It enables "1-Click automatic well design", where all planning is done within a single page in any modern web browser (Hassan et al., 2022). The software will, for instance, automatically generate well trajectories and casing design. Moreover, the software is updated in real time with the latest changes, improving work flow significantly. The software does, however, not address the operational- and production phases of the well as discussed in the PhD-design (Brechan, 2020).

2.1.2 Experience Transfer

Using experience from previous operations is key and described as one of the two most critical stages in well planning (Tvedt, 2017). It facilitates use of proven engineering work and design and saves time. Due to its importance, having an effective way of storing and utilizing previous experience is crucial. Today, experience transfer is a human-based process. Most of the experiences are noted in text-based documents stored in large databases containing thousands of documents (Brechan, 2020). Some of the documents are used when establishing new well plans or when governing documentation and best practices are updated. However, a large number of them sit untouched. Experiences must be searched for using specific keywords. The process is heavily dependent on the personnel extracting the data, as depicted in Figure 2.3. Making use of correct and relevant experience can therefore be hard, specially for inexperienced engineers. Moving from human-oriented to data-oriented experience transfer will be important to meet the constantly higher standards in terms of cost, effectiveness and emissions. Data-oriented experience transfer require software that automatically links previous experiences with future operations.

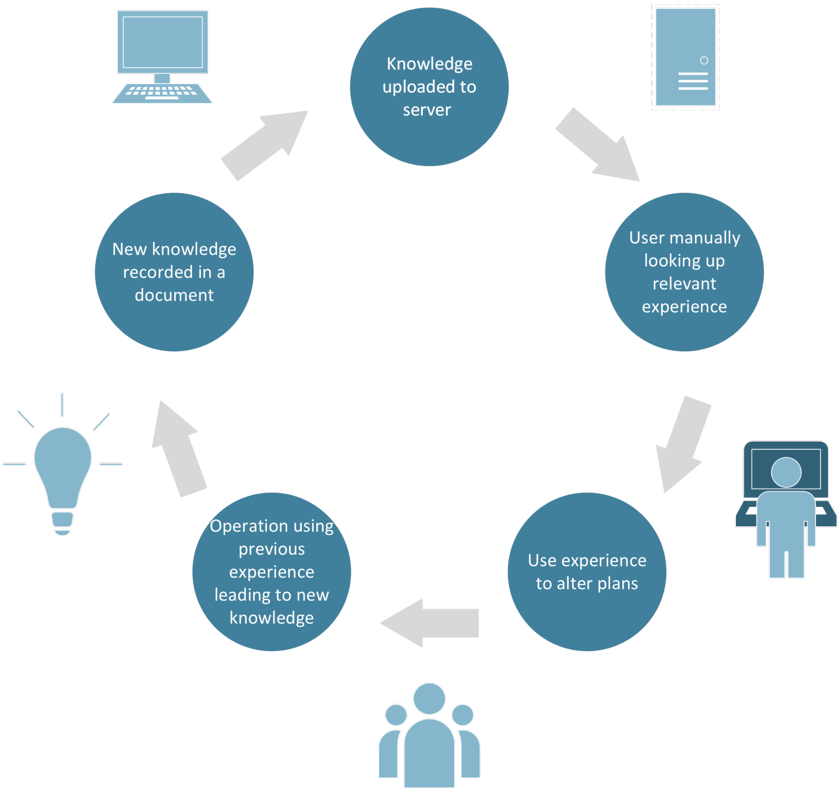


Figure 2.3: Experience transfer in today's industry.

Digital experience should and can be readily available for all that wishes to incorporate it within their plans. The PhD by Brechan (2020) has already presented a method for how experience can be automatically incorporated to plans without the need of searching through countless non-related documents. Once experience has been used in the planning phase of the well, the system knows which documents that have been used, and can save it for use later in the life cycle of the well. This can also help other disciplines work more efficiently at a later stage of the well's life, such as interventions.

A well goes through multiple phases throughout its life. The life cycle of a well contains all phases from planning to drilling, through production and all the way to plugging and final abandonment.

A typical well can have a lifespan of about 20-30 years. It is therefore crucial to implement digital experience that can be stored and structured in a way readable by both man and machine. As well planning and experience transfer stands in the industry today, value is lost due to the fact that today's systems are sub-optimal. By implementing experience from neighbouring wells and other relevant wells in planning, drilling efficiency, safety and overall well quality is improved.

2.1.3 Digital governing documents

As with experience, digitalization and automation can lead to better handling of governing documents than today. Governing documents are a set of regulations, acts and standards that has to be followed in the industry. Governing documents are fixed in a hierarchy, see Figure 2.4.

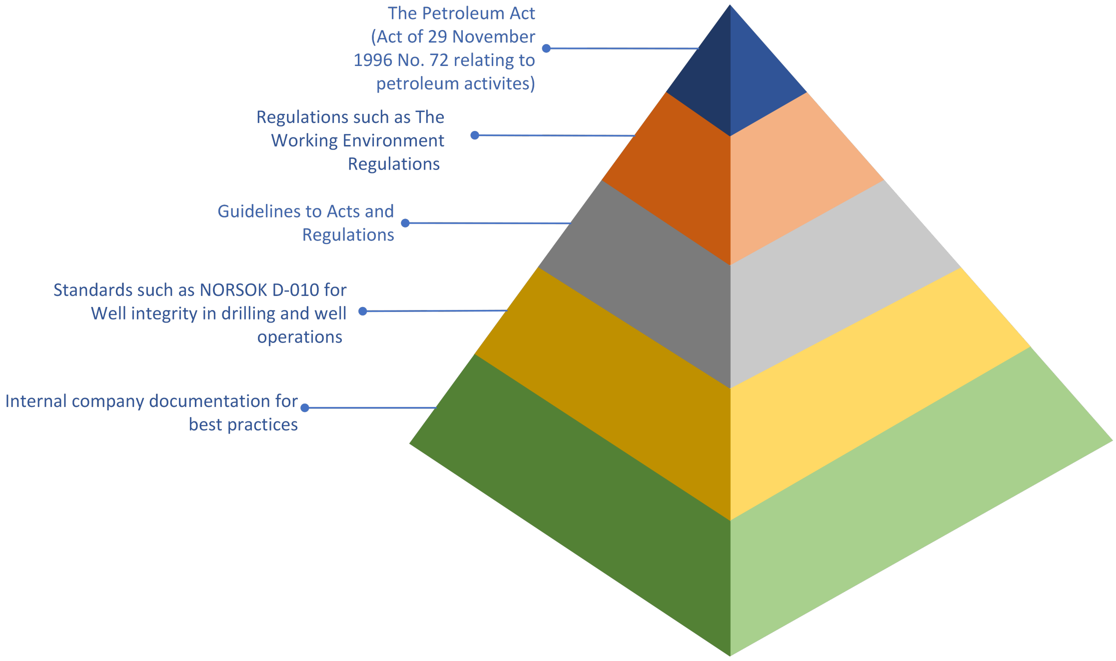


Figure 2.4: Hierarchy of governing documentation.

In the Norwegian sector, acts and regulations such as the Petroleum Act (Act of 29 November 1996 No. 72 relating to petroleum activities) serves as the general legal basis for petroleum activities. In the hierarchy below the Petroleum Act, there are regulations regarding working conditions etc., as well as guidelines to the acts and regulations. The three upper layers of the pyramid are typically general, and does not provide details and specific technical restrictions. The fourth layer, on the other hand, does. This consists of standards such as NORSOK D-010 for well integrity in drilling and well operations. NORSOK D-010 is a standard and set of best practices built from experience since the start of the petroleum industry in Norway. NORSOK D-010 was first released in 1994, and it serves to ensure a safe, reliable and efficient way of drilling, completing and plugging wells. On the final layer of the hierarchy pyramid there are internal company documents which provides guidelines and best practices for operations.

The Petroleum Safety Authority Norway (PSA) is responsible for ensuring that companies operate within legal boundaries on the Norwegian continental shelf (NCS). There is a difference between the acts and regulations, and industry standards and internal company standards. However, companies are expected to operate within the recommendations that for instance NORSOK D-010 provides. If not, the PSA might investigate depending on the situation. For this purpose one of the main changes in NORSOK D-010 when revision 5 was released in 2021, was the clear distinction between the words "shall", "should", "can" and "may". Shall is listed in NORSOK as a requirement, where no deviation from the standard may apply, whereas this does not apply for the rest. A clear line between what is required and what is recommended also helps with digitalization and automation, as it gives an unambiguous instruction to what is tolerated and not. As with experience handling, building an automated system for governing documents is crucial for safe and efficient operations.

2.1.4 Automation

Many industries have improved their workflow and processes by the use of automation over the last decades. The petroleum industry has also started adopting automated solutions. However, there is still a potential for further improvements such as within well planning and drilling.

Currently, no company offer a software that fully automate the well planning process. However, there are solutions that have automated parts of the process. The most advanced software with regards to this is perhaps Oliasoft's WellDesign, as mentioned in Section 2.1.1. 10% of effective operating time and 30% of non-productive time (NPT) can be saved using an automated planning solution with appropriate incorporation of experience (Brechan, 2020). The saved time consumption (man hours) contributes, but the main savings come from a digital platform that includes experiences and produce high quality plans.

Drilling automation is an area that is gathering momentum and becoming more widely used. There is widespread consensus that it will play an important role in the industry going forward. The Drilling Systems Automation Roadmap is an initiative started to accelerate adoption of advancements in drilling automation systems. They state the following in their 2025 vision: "Well plans are uploaded into an interoperable drilling system that automatically delivers a quality wellbore into the best geological location" (De Wardt et al., 2015), as cited in (Brechan, 2020, p. 12). Further investments must be made to reach this level of automation. Ziatdinov et al. (2021) point at three specific reasons to why investments should be made within the field of drilling automation:

- Automation of drilling moves people out of danger zone, improving personal safety.
- Execution is transferred from human to machine, reducing the risk of human error.
- Drilling becomes more efficient as operations are more consistent and use of self-learning algorithms allow each successive well to be drilled better than the previous.

Field trials also suggest that drilling automation improve performance. Already in the first deployment of an automated drilling control system, Statoil (now Equinor) saved 10% rig time per well that was drilled (Abrahamsen et al., 2015). 4% of these came from automating repetitive tasks such as tripping and pump start-ups, while the rest came through safeguarding and optimising manual operations. Additionally, even though the system was designed to eliminate human errors and major accidents, the trials showed that invisible lost time was reduced by avoiding minor issues that normally would slow down the operation (Abrahamsen et al., 2015).

Similarly, a case study on NOV's automated drilling system NOVOS and wired drill pipe, demonstrates that drilling efficiency is improved (NOV, 2019). Specifically, rate of penetration (ROP) were increased while both cost per foot and mean square error versus offset were reduced using the system.

In the Middle East, several automation tests on onshore gas wells were performed over a four year period. Over 60.000 feet was drilled using an automated system, focusing on fully automating drilling a stand. Several key performance indicators (KPIs) were recorded in the process, leading to the conclusion that ROP was increased by 20%, and no safety issues were recorded (Gomez et al., 2021). A clear distinction between degrees of autonomy in drilling is also given in this paper, as seen in Figure 2.5.

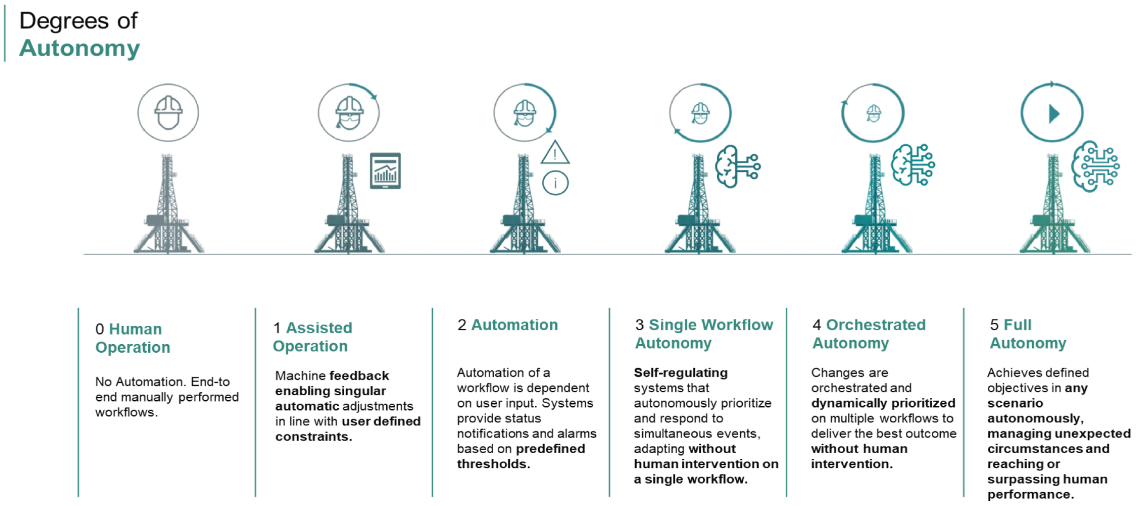


Figure 2.5: Degrees of autonomy (Gomez et al., 2021).

An analysis of performance data from a North Sea offshore field also reveal large discrepancies in production output from different control room crews. More than 5 % difference were detected between different crews (Brun et al., 2017). A 12 % difference were also found at another asset, emphasising the human factor. Automation is a key enabler for minimizing human influence on efficiency and production output.

Moreover, 80% of all accidents on offshore facilities in the United States can be directly linked to human error (Whitfield, 2017). This aspect of improved safety, as well as the other examples indicating that operations are more efficient and consistent using drilling automation, clearly show why full autonomy should be implemented on drilling rigs.

2.1.5 Rig Automation Software (RAS)

To fully utilize drilling automation, it is important that well plans are made available in a format compatible with the RAS available, meaning that full well plans can be fed directly to and understood by the RAS. The plan can then be executed by the system on the rig under surveillance from the drillers. This was also mentioned as one of the main objectives of this thesis. Different software are connected to different equipment and require different input. Two examples of current RAS are presented in Table 2.1. Regardless of system, each event planned to be performed in the well must be represented by and translated to readable input/instructions for the RAS. In this thesis, a simple approach is used to illustrate how a given well procedure can be represented in a way readable for RAS. This is elaborated on in Section 3.4.3.

Table 2.1: Rig automation software. Readers are encouraged to visit the companies web pages for more information about the software systems.

NOVOS	NOVOS is a reflexive drilling system, meaning that it starts performing a set of tasks when prompted (NOV, n.d.). It is the most deployed drilling automation system in the world. The system automates repetitive drilling activities while the driller observes. The driller can at all times take over control over the system. NOVOS uses an open application platform allowing the software to be tailored to the needs of the user.
HMH	HMH provide automated drilling solution called drillersAssist. It is a smart module that assists the driller with automation of well construction sequences (HMHW, n.d.). Also here, the main intentions are to automate repetitive drilling activities as well as making operations safer and more efficient.

2.1.6 Benefits of digital well planning

Significant changes has not been seen in tools for planning and constructing wells over the last few decades. Over the coming decade, however, considerable changes in well planning is expected (Brechan, 2020). Key benefits of moving to a fully developed automated well planning solution can be summarized as the following (numbers are not indicative of importance) (Brechan, 2020):

1. Improved work flow.
2. Automation of repetitive tasks.
3. Improved experience handling.
4. Automatic verification of compliance with governing documentation.
5. Less dependency on human interaction.
6. Digital programs readable for man and machine.
7. Governing documentation and standards.
8. Less human influence.

1: Barriers between compartments will be minimized. Data will flow more freely between different contributors and sources in the planning process. Software is updated in real time, allowing all parts involved to be up to date on latest changes.

2: Time spent on repetitive task will be reduced significantly. Automation reduces risk of mistakes in planning phase as well as overall workload. Best practices facilitate high efficiency.

3: Experience handling will be improved as relevant experiences can be incorporated into new well plans automatically i.e. by the software itself. No need for looking up experiences using key words and not dependent on people involved. Experience is made by humans, but implemented to the automatically developed plan. System ”learns” from previous operations/plans.

4: Engineering is run automatically and well plans are automatically checked against governing documentation and other requirements. Feedback is provided and options to alter plan given.

5: Less human interaction needed in planning process. System is ”smart” and suggests possible solutions etc. Risk of mistakes decreases.

6: Programs are made readable for man and for RAS, enabling automated, safer and more effective operations.

7: Governing documentation and standards are implemented in software automatically integrating it to the plans.

8: Individual influence from humans will be reduced as many processes are standardized and done automatically based on previous experiences.

2.1.7 Benefits by digitalization in the petroleum industry

Digitalization in the petroleum industry can trigger significant savings. Rystad Energy released an in-depth study of savings in the petroleum industry in 2019 (Rystad Energy, 2019). The study suggests that by the end of the 2020's, as much as $100 billion can be saved from upstream budgets by implementation of automation and digitalization. Service companies are reinventing themselves in order to aid operators in attaining these savings, according to the study. Figure 2.6 shows digitalization and automation potential by budget cost.

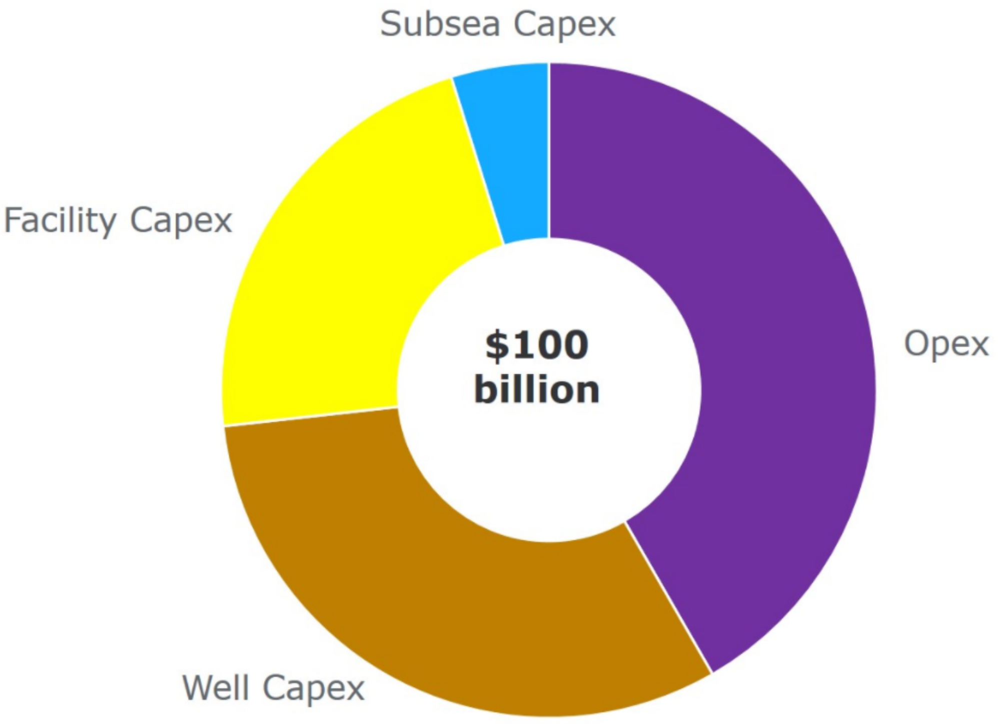


Figure 2.6: Digitalization and automation potential by budget cost (Rystad Energy, 2019).

In 2018, across more than 3000 companies in the upstream segment of the industry, $1 trillion was spent on operational expenditure (OPEX), wells, facilities and capital expenditures (CAPEX) (Rystad Energy, 2019). By implementing digitalization and automation and using these effectively, around 10% can be saved. The savings can even become more significant and especially for drilling costs and subsea facility costs. These savings can be towards 20% and 30%, respectively. Across the industry there are however a broad specter of different companies and fields, from larger national oil companies to smaller exploration and production (E&P) companies. The realistic savings across the entire upstream segment of the industry will therefore be closer to 10%, due to how the different companies can implement new technologies (Rystad Energy, 2019).

Reducing NPT in drilling is a major driving force for further digitalization of the petroleum industry. NPT is the down time during a drilling operation and can result in significant additional costs. For many decades the average NPT has been 10-15%, depending on project complexity (Pritchard et al., 2012) as cited in (Brechan, 2020). Many efforts have been made to reduce NPT. As mentioned in Section 2.1.4, field trials with automated rigs have proven to significantly improve performance. Moving from a text-based to fully digital well planning, can reduce NPT with 30% and give 10% more effective operations (Brechan, 2020), both accounting for significant savings.

Savings can be made in numerous areas within the industry using digitalization. Çağlayan Arkan, Microsoft's Vice President of Manufacturing & Supply Chain, together with companies like DNV GL, Emerson, Equinor and Siemens, identified so-called key digital opportunities in the industry which could account for billions of dollars in savings (Arkan, 2018). These are shown in Figure 2.7 and are important for further efficiency improvements.

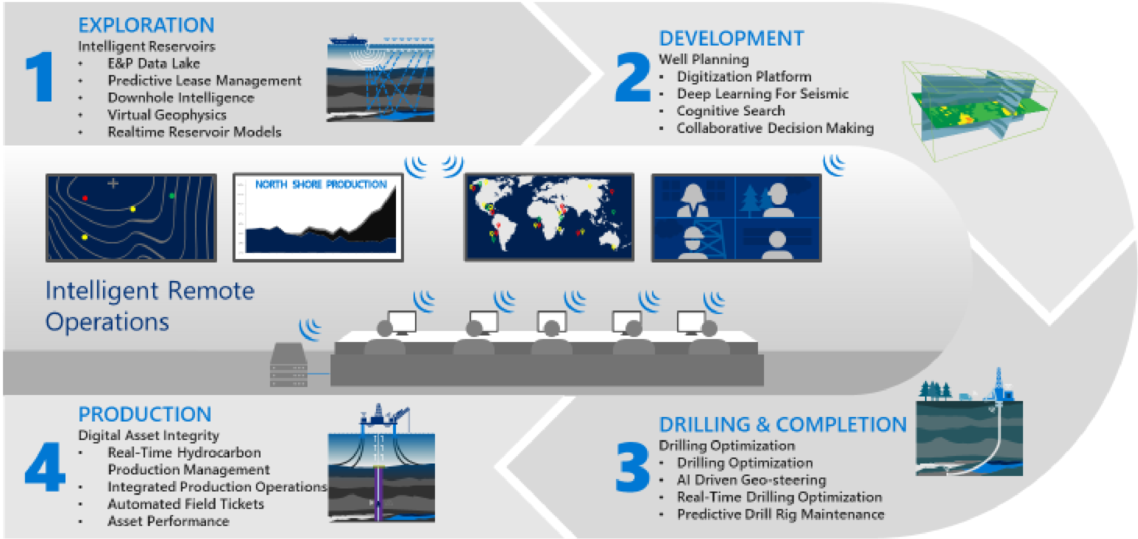


Figure 2.7: Digital hotspots identified by Microsoft (Arkan, 2018).

Particularly artificial intelligence (AI) is highlighted as an area that could bring high value if implemented. This requires analysis and utilisation of vast amounts of data. The petroleum industry has been collection enormous amounts of data for many years already, however it has been poor at utilizing it. Specifically, less than 1 % of the data at hand is being used (Brun et al., 2017). The potential for better utilization of the data available is clear. Back in 2014, consultants from the renowned Bain & Company, stated that production could be increased by 6 to 8 % by improving the use of data analytics (Padmanabhan, 2014).

2.2 Well Operative System (WOS)

As mentioned in the introduction, the research presented in this thesis builds on the work in Brechan (2020), which is a blueprint of a LCWIM. This is a model for planning and construction of wells made to ensure well integrity throughout the life cycle of the well. The LCWIM contains functions to handle administration, well engineering and production of programs and procedures, as well as all things listed in Section 1.1. Digital well management (DWM) is the name of this automated process of well planning. The "brain" in this model, the so-called well operative system (WOS), enables DWM. Many of the principles presented in the WOS have been applied in the WPS presented later in this thesis. Figure 2.8 is made to show how the WPS and DDOP are relative to DWM. The WPS is equivalent to the WOS in the LCWIM, while the DDOP is a sub-level of the event manager in the LCWIM. The event manager connects all elements of the WOS, and serves as the hub for program information.

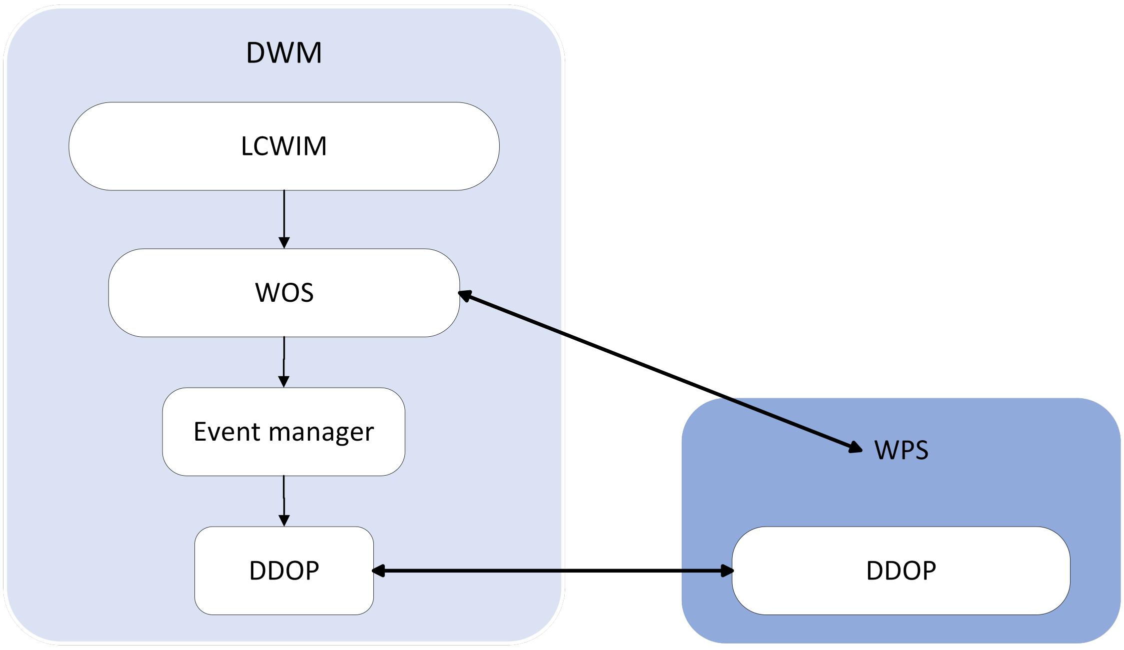


Figure 2.8: WPS relative to DWM.

The WOS is a digital framework designed to tie together all disciplines in well planning in a digital and automated manner (Brechan, 2020). This system is differentiated from other automated engineering software currently on the market, as it also provides automated support for administrative tasks and provides digital programs and procedures. The system enables a smooth work flow, as any task from engineering to administrative work for each well is designed to be done by the software. It also provides all relevant personnel access to updated information about the well. Figure 2.9 and Figure 2.10 illustrates how well planning is done today and how it can be done using the WOS. As can be seen, today's process is a iterative and time consuming process. By using an integrated software such as the WOS, it is estimated that a significant amount of man hours will be reduced and that total well cost cut with 20 to 30% due to automated incorporation of experience (Brechan, 2020).

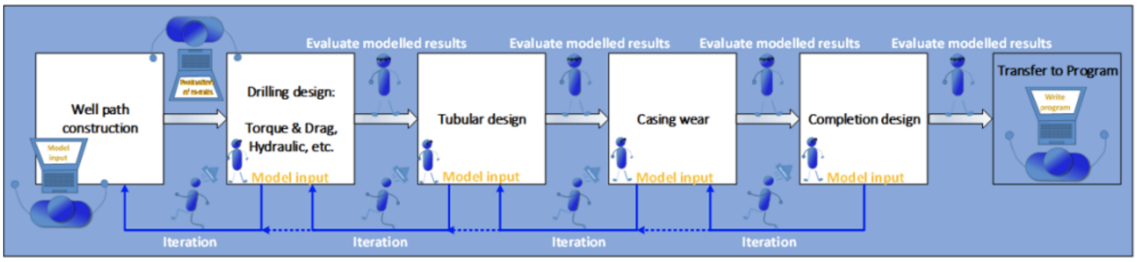


Figure 2.9: Well planning today (Brechan, 2020).

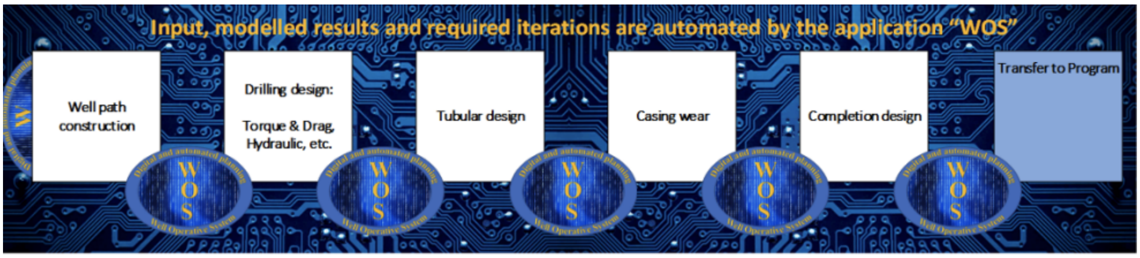


Figure 2.10: Fully automated well planning (Brechan, 2020).

In the LCWIM, the WOS gathers relevant data from subsurface models. This is distributed to different modules to initiate development of a digital program for a given objective (drill section, set casing, etc.) in the well, see Figure 2.11. As the figure shows, in intervention mode these modules can be wellpath, activity design, casing/tubing design and casing wear. In short, the WOS makes an initial program (well plan) using this data. The software then revise the plan using digital experiences and best practices implemented in the system. Additionally, a loop is run to tune operational parameters to the optimum. Finally, the WOS gives feedback to how good the final program is.

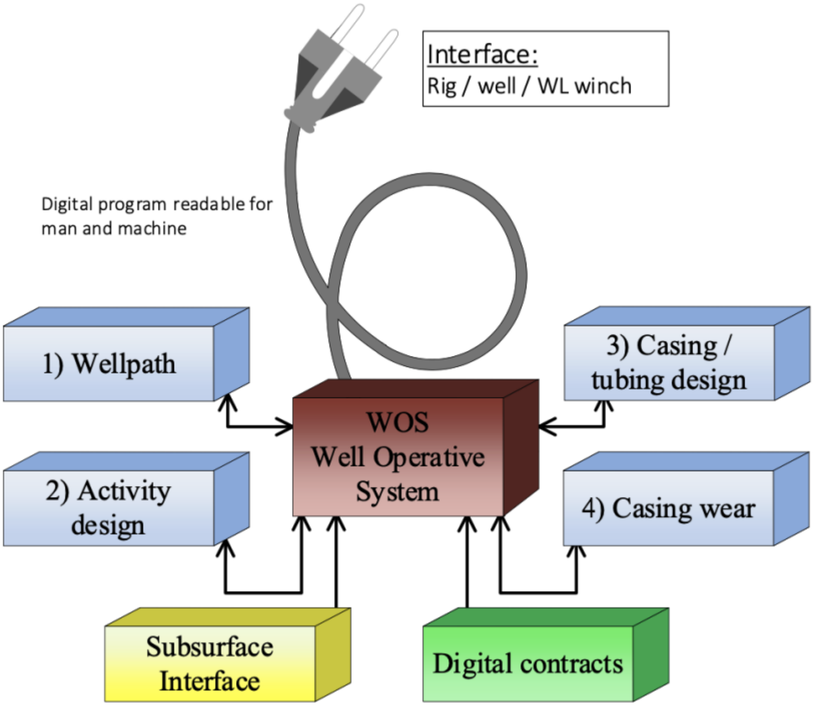


Figure 2.11: LCWIM overview in intervention mode (Brechan, 2020).

On a more detailed level, Figure 2.12 shows an outline of how the first iteration performed by the program is, and how the different modules in the LCWIM are connected. An initial wellpath is proposed based on subsurface data and well head (WH)- and target coordinates. Furthermore, section- and casing program is found using analysis of pore pressure plots and mud weights from bottom to top (Brechan, 2020). Predefined BHA's are selected based on known hole size, casing program and tortuosity of the wellpath. Initial engineering calculations are performed and operational parameters mapped with minimum and maximum value to ensure future validation in the program (Brechan, 2020). Also standard equipment and services from contracts are integrated into the well plan at this point.

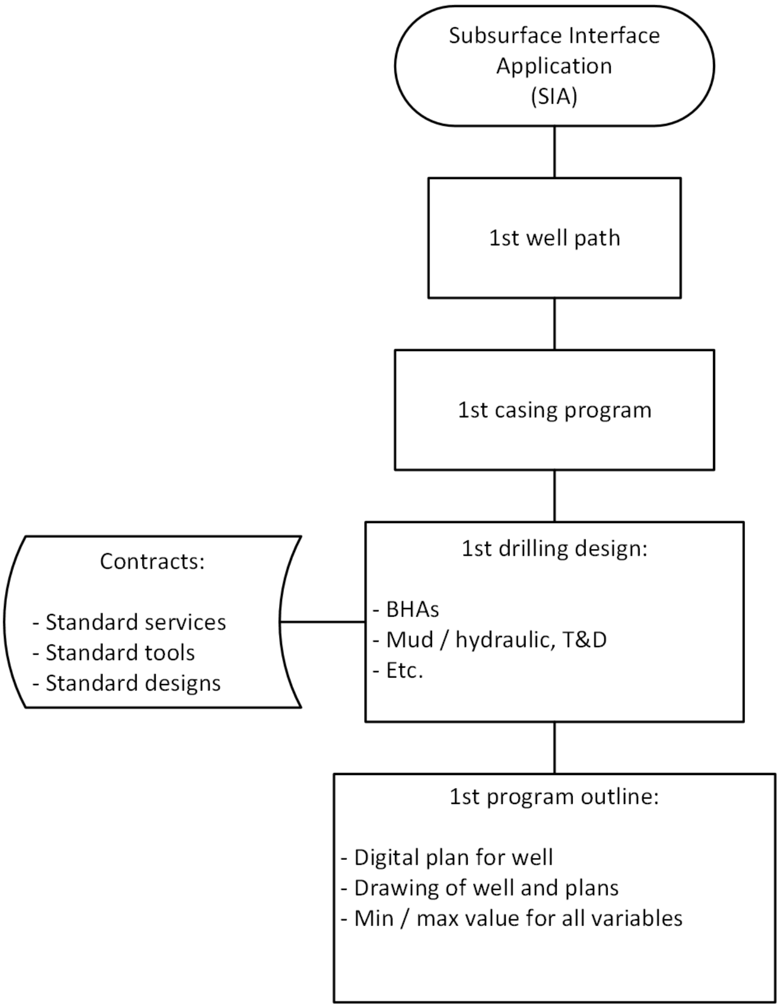


Figure 2.12: Flow chart for establishment of first well plan in WOS (Brechan, 2020).

The WOS then runs the initial well plan through the different modules and functions in the LCWIM in an iteration sequence. Here, engineering and parameters are tuned to improve the quality of the plan and ensure its validity. Also experience from similar plans and operations is designed to be incorporated into the plan at this stage. The iteration sequence illustrated in Figure 2.13 shows how the lines in the event manager are created by the WOS. The sequence is also paraphrased from Brechan (2020, p. 71) below. Important terminology used in the iteration sequence is described in Table 2.2.

Table 2.2: Terminology used to describe the iteration sequence.

Terminology	**Description**
Event manager	Main hub for engineering design. It is also the main hub for man and machine to read all events necessary to achieve the desired well design.
Control system	System for controlling engineering calculations. Makes sure that all variables are within limits and that other regulations are met.
Variable map	Map that holds current-, minimum- and maximum values of all parameters.
Digital experience	Experience from previous planning and operations in a digital format that can be incorporated into new plans.

The process is designed as follows:

1. The input is fed into a control system by the event manager. The variables are then reviewed to see if they have changed significantly enough to require recalculation of the engineering.
2. The variable map is linked to the control system. The variable map includes all values currently impacted by the control system, while also tagging for additional engineering computations influenced by the control system. It also includes the allowable minimum and maximum values per engineering.
3. Any engineering calculations that use the variable will change from "green" to "yellow" if the variable has to be modified.
4. The method manager is in charge of selecting new tools or adjusting existing parameters if a calculation falls outside of the acceptable range. After that, the engineering will be performed once more to determine new minimum and maximum values.
5. The engineering loop will begin if all of the variables from step 3 are within acceptable ranges:
    (a) Internal priorities are established, and calculations are completed one at a time.
    (b) The control system determines whether the updated variables have changed more than the recalculation lower limit. If not, the previous value is kept.
    (c) If all variables are between the set minimum and maximum values, the loop returns to step 5a, where the remaining engineering calculations are completed in the given priority. If this is not the case, the loop terminates and returns to step 4. The loop finishes once all of the calculations have been completed and are within an acceptable range.
6. The next stage is to add experience using the digital experience module; experience can be included or ignored depending on whether or not it clashes with engineering. After adding experience, a digital program (or one line in the event manager) is created, and the iteration process may restart.

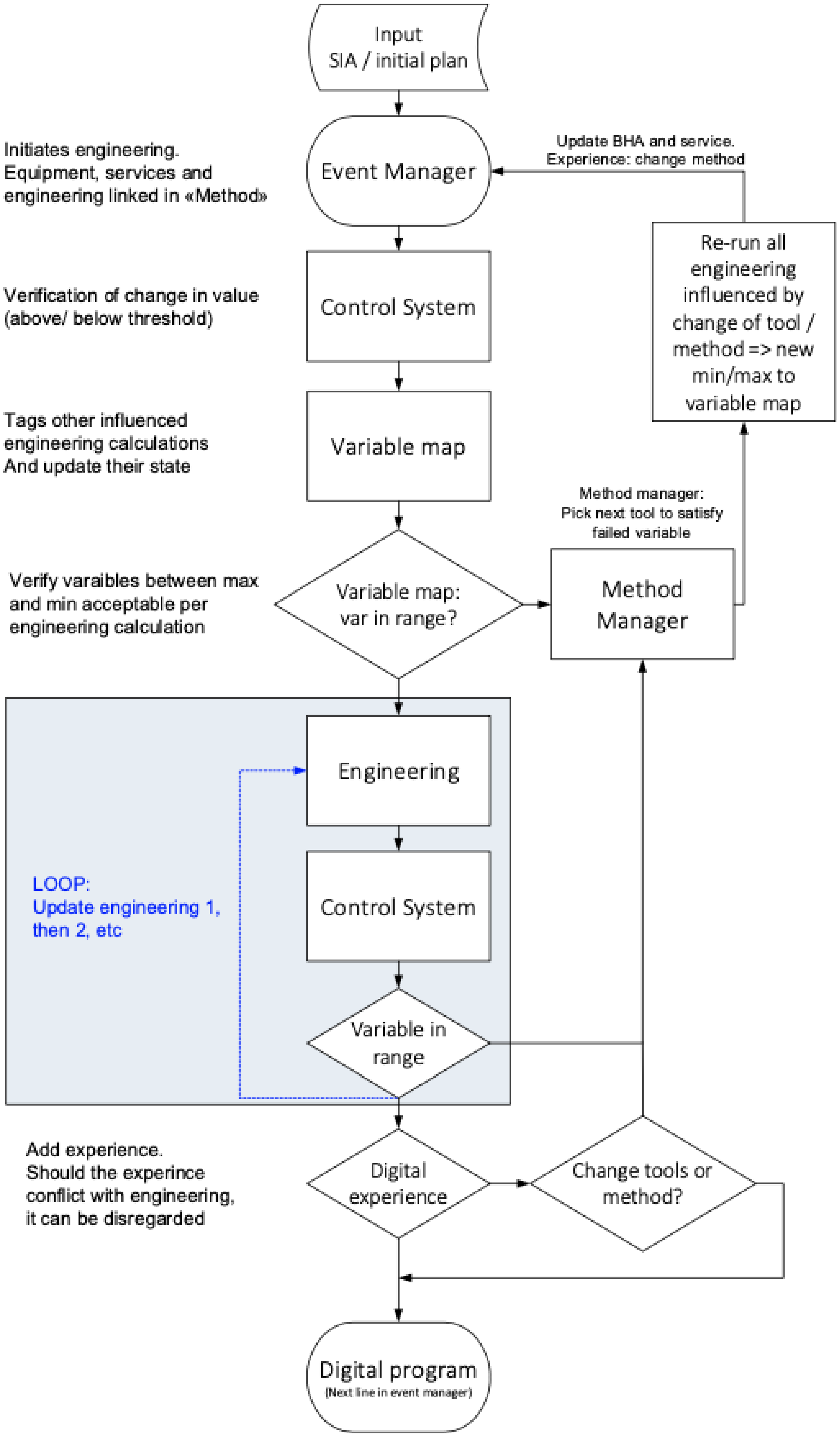


Figure 2.13: Iteration sequence in the LCWIM made by the WOS (Brechan, 2020).

When the iteration sequence has run through all events in the well plan, the final plan is ready. The WOS creates and tunes the event manager, which is like a time-planner or plan for the operation readable for both man and machine. This is an important feature to facilitate automated operations in the well. For more in-depth information about the WOS, the authors refer to Brechan (2020).

2.3 Programming

The various programming methods and concepts used in this thesis are grouped into:

- Databases
- OOP
- Python programming language

A brief intro follows below. For a full introduction to these concepts please see Appendix A, Appendix B and Appendix C.

2.3.1 Databases

A database is an organized collection of structured information, or data, typically stored in a computer system (Oracle, n.d.). There are two different types of databases that are used today, relational and non-relational databases, with relational being the most popular. Relational databases use a structured query language (SQL) to query, manipulate and defining data within databases (Oracle, n.d.). There are plenty of different databases to choose from, but one of the simplest for prototyping and development is SQLite3. SQLite3 was therefore used in the WPS. For more information about databases see Appendix A.

Specifically, databases are used to save the DDOPs created in the WPS. The DDOPs are made up of several rows containing various data, making it favorable to store it in a database. Using databases for this purpose also enables agile planning as the DDOP can be adjusted using simple commands.

2.3.2 Object oriented programming (OOP)

Object oriented programming is a widely known programming approach used today. It uses classes and objects to create programs and is based on four main principles: inheritance, encapsulation, abstraction and polymorphism. These principles provide code that is recognized by being reusable, safe and easy to debug. How OOP is used to develop the WPS is further explained in Section 3. For more information about OOP, see Appendix B.

2.3.3 Python

Python is a widely used high level programming language. The language is used for different purposes such as web and software design, data analytics and data visualization (Future Learn, 2021). This versatility was important to why Python was chosen as the programming language early in the development phase. Specific programming methods and techniques were yet to be decided at this point and Python offered the flexibility needed to shape the software as wanted. Additionally, Python has several libraries and packages available. These contain simple and complex functionality that can be used in programs and was one of the main reasons to why this language was chosen.

Appendix C gives an introduction to Python, as well as each package used in the development of the WPS. Table 2.3 gives an overview of how the different packages are applied in the WPS.

Table 2.3: Python packages and how they have been applied in the WPS.

Python package	**Relevance for WPS**
SQLite3	Database for storing DDOP.
Numpy	Engineering calculations.
Pandas	Importing Excel files.
Matplotlib	Plotting of wellpaths.
Pillow	Importing images shown in GUI.
CSV	File format for automation compatible DDOP.
Tkinter	Visual effects GUI.

2.4 Engineering

A well planning application requires implementation of engineering calculations and functionality. In the application presented in this thesis, a few calculations are implemented to demonstrate the link between the WPS, WOS and the LCWIM. These do per date only cover parts of hydraulic engineering calculations, as the calculations in themselves are not part of the main objective of the thesis. The implemented calculations illustrate how the engineering can be directly tied to each step in the well plan, i.e. each line in the event manager, and provide feedback to influence the plan. It is possible to include all engineering using the same approach.

2.4.1 Pressure loss

It is important to maintain control of the pressure in the well. If not, one may experience problems such as fracturing of the formation, getting kicks or even blowouts. When drilling and circulating fluid, pressure losses will occur and can be the cause of the mentioned problems. The hydraulic system is depicted in Figure 2.14. Many approaches can be used to calculate pressure loss. In this thesis, the Power Law model is used and will be presented (Bourgoyne et al., 1986).

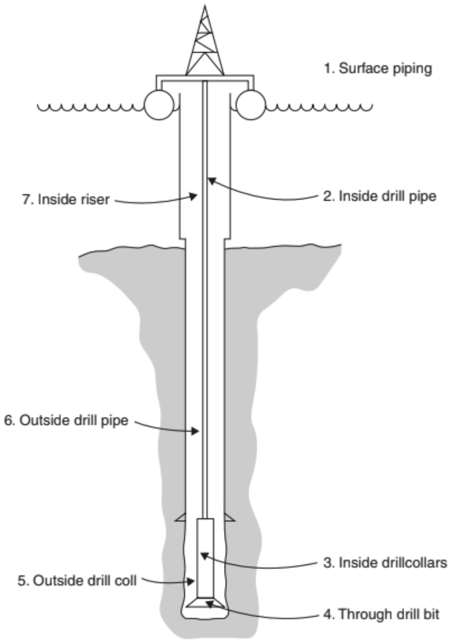


Figure 2.14: The hydraulic system (Aadnoy, 2010).

The average velocity inside the drill string is given by Equation 2.1.

$$\overline{v}_{pipe} = \frac{q}{2.448 d_1^2} \tag{2.1}$$

Similarly, the average velocity in the annular space is given by Equation 2.2.

$$\overline{v}_{annulus} = \frac{q}{2.448(d_3^2 - d_2^2)} \tag{2.2}$$

Flow behavioral parameters n and K are given by Equation 2.3 and Equation 2.4.

$$n = 3.32 \log \frac{\theta_{600}}{\theta_{300}} \tag{2.3}$$

$$K = \frac{510\theta_{300}}{511^n} \tag{2.4}$$

The pressure loss will depend on the flow regime present. Thus, the critical Reynold's number is calculated for both inside and outside the pipe, as given in Equation 2.5 and Equation 2.6, respectively.

$$N_{Re,pipe} = \frac{89100\rho\overline{v}^{2-n}}{K}(\frac{0.0416 d_1}{3 + \frac{1}{n}})^n \tag{2.5}$$

$$N_{Re,annulus} = \frac{109000\rho\overline{v}^{2-n}}{K}(\frac{0.0208(d_3 - d_2)}{2 + \frac{1}{n}})^n \tag{2.6}$$

If the critical Reynold's number is below a threshold, typically 2100, the flow is laminar. The pressure loss both inside the pipe and in the annular area are then defined by Equation 2.7 and

Equation 2.8, respectively.

$$\Delta P_{lam.,pipe} = \frac{K\overline{v}^n L}{144000 d_1^{n+1}}(\frac{3+\frac{1}{n}}{0.0416})^n \tag{2.7}$$

$$\Delta P_{lam.,annulus} = \frac{K\overline{v}^n L}{144000(d_3-d_2)^{n+1}}(\frac{2+\frac{1}{n}}{0.0208})^n \tag{2.8}$$

Conversely, if the flow is turbulent, the pressure losses inside pipe and in annulus are defined by Equation 2.9 and Equation 2.10, respectively.

$$\Delta P_{turb.,pipe} = \frac{f\rho\overline{v}^2 L}{25.8 d_1} \tag{2.9}$$

$$\Delta P_{turb.,annulus} = \frac{f\rho\overline{v}^2 L}{21.1(d_3-d_2)} \tag{2.10}$$

Pressure loss through the bit is given by Equation 2.11.

$$\Delta P_{bit} = \frac{8.311 * 10^{-5}\rho q^2}{C_d^2 A_t^2} \tag{2.11}$$

Where:

$\overline{v}$ is mean velocity [ft/s]

q is flow rate [gal/min]

d_1 is internal diameter (ID) of pipe/collar [in]

d_2 is outer diameter (OD) pipe/collar [in]

d_3 is hole diameter [in]

θ_{300} is Fann reading at 300 RPM [-]

θ_{600} is Fann reading at 600 RPM [-]

K is consistency index [-]

N_{Rec} is critical Reynold's number [-]

n is flow behavior index [-]

ΔP is pressure loss [psi]

f is friction factor [-]

L is pipe length [ft]

ρ is density [lbm/gal]

C_d is nozzle coefficient [-]

A_t is total nozzle area [in^2]

Pressure loss in surface equipment, as per Figure 2.14, is not included here.

2.4.2 Surge & swab

Additional pressures occur when the drill string moves up and down in the well. Movements can be caused by tripping or by the heave motion of the rig. The additional pressures, both negative and positive, are called surge & swab pressures and must be accounted for when planning a well. If not, the pressure in the well may exceed the upper and lower limits and cause problems.

The approach used to calculate surge & swab pressures in this thesis is presented below (Lapeyrouse, 2002).

Step 1: Determine n as per Equation 2.3.

Step 2: Determine K as per Equation 2.4.

Step 3: Determine fluid velocity using Equation 2.12 or Equation 2.13.

For plugged flow:

$$v = \left[0.45 + \frac{d_2^2}{d_3^2 - d_2^2} \right] v_p \tag{2.12}$$

For open pipe:

$$v = \left[0.45 + \frac{d_2^2 - d_1^2}{d_3^2 - d_2^2 + d_1^2} \right] v_p \tag{2.13}$$

Step 4: Find maximum pipe velocity, as per Equation 2.14.

$$v_m = 1.5 \times v \tag{2.14}$$

Step 5: Determine pressure change using Equation 2.15.

$$\Delta P_{surge/swab} = \left(\frac{2.4 v_m}{d_3 - d_2} \times \frac{2n+1}{3n} \right)^n \times \frac{KL}{300(d_2 - d_1)} \tag{2.15}$$

Where:

v is fluid velocity [ft/min]

v_p is velocity of pipe [ft/min]

v_m is maximum pipe velocity [ft/min]

$\Delta P_{surge/swab}$ is pressure change [psi]

2.4.3 Hole cleaning

Hole cleaning is an important aspect of drilling a well. Cleaning is influenced by mud properties, cuttings size, flow rate, hole deviation and more. In this thesis, the carrying capacity index (CCI) is calculated as shown below to check if hole cleaning is done satisfactory (ASME Shale Shaker Committee, 2004).

Using Equation 2.2 (changed to ft/min), Equation 2.3 and Equation 2.4, the CCI can be found using Equation 2.16.

$$CCI = \frac{K \times v_{ann} \times MW}{400000} \tag{2.16}$$

Where:

CCI is hole cleaning index [-]

v_{ann} is annular velocity [ft/min]

MW is mud weight [ppg]

2.4.4 Engineering failure criterions

Standards and governing documentation will dictate the frame for operations and well plans. Criterions must be defined to know when engineering fail. The criterions may be set differently for each country, field or well type, depending on conditions, companies involved, local regulations and experience.

The LCWIM design for controlling parameters comprises the ”variable map” and ”control system”, which was discussed with Figure 2.13 in Section 2.2.

2.5 Digital Detailed Operational Procedure (DDOP)

This is the WPS' hub for running through the planned events and derive an optimal combination of operational parameters and engineering variables. The DDOP has the same level of detail as DOPs used in the industry today. A part of the DDOP is even more detailed. This is the part intended to communicate with RAS, see Section 2.1.5 for more.

The DDOP in this thesis was developed in the autumn project, and it covers operations for drilling a 17 1/2” section. This DDOP can be considered ”high level” digital experience and a copy of what is ”company best practice” for drilling 17 1/2” sections. In the LCWIM, the full DDOP will be incorporated early in the automated planning sequence, as discussed with Figure 2.12 and Figure 2.13. Once the planning database has been populated with digital experience as shown in Figure A.1, parts of the DDOP and lines in the event manager will be changed automatically in the planning.

The DDOP format serves two purposes. Primarily, it builds on an existing system used to describe all activities in well construction, intervention and P&A. This is the rig activity reports. This basis lets all industry professionals world wide recognize the text and operate the software for planning. Detailing every single rig activity and linking all info together enables the secondary purpose, which is a digital format understandable for other software such as rig automation and experience transfer. The understandable format can easily be applied to form experiences specific to certain conditions, which in turn automatically is incorporated in subsequent planning. This provides a seamless and fully integrated loop for planning and operations. The digital format of planning wells can also keep operational history, and it forms the basis of a dynamic planning tool.

3 Methodology

One of the main objectives with this thesis was to make a working prototype of the WOS and prove central design theories for the WOS and how it is intended to operate within the LCWIM. To solve this, the WPS was developed. In this section, the architecture of the WPS is explained. The application itself is presented in Section 4.

3.1 Project Scope

A full well planning application integrating all parts of planning is comprehensive and development of all necessary modules is an extensive job. Defining the project scope is therefore important. The thesis' project scope can be defined as proof of concept for a well planning software integrating engineering, contracts, equipment and automated rig equipment. All of these are important concepts in the LCWIM. Some additional functionality is integrated in the WPS compared to the project scope. Connecting the different parts and making them work together will be the main focus. It is emphasized that the goal of this thesis is not to create a full well planning application.

3.2 Application Overview

The WPS consists of two main parts; the main menu and the planner window, see blue boxes in Figure 3.1. In the main menu, the user can enter input which is later used in the planner window. Additionally, the main menu has functionality to show different plots of the chosen wellpath.

The other main part of the WPS is the planner window. The primary functionality of the planner window is to create well plans. This is where the actual well plan (DDOP) is made by defining all events necessary to reach the desired objective. This is elaborated on in Section 3.4.1. The DDOP has two connected modules: engineering and automation. The engineering module runs calculations for all events in the well plan, while the automation module translates the DDOP to a format compatible with RAS.

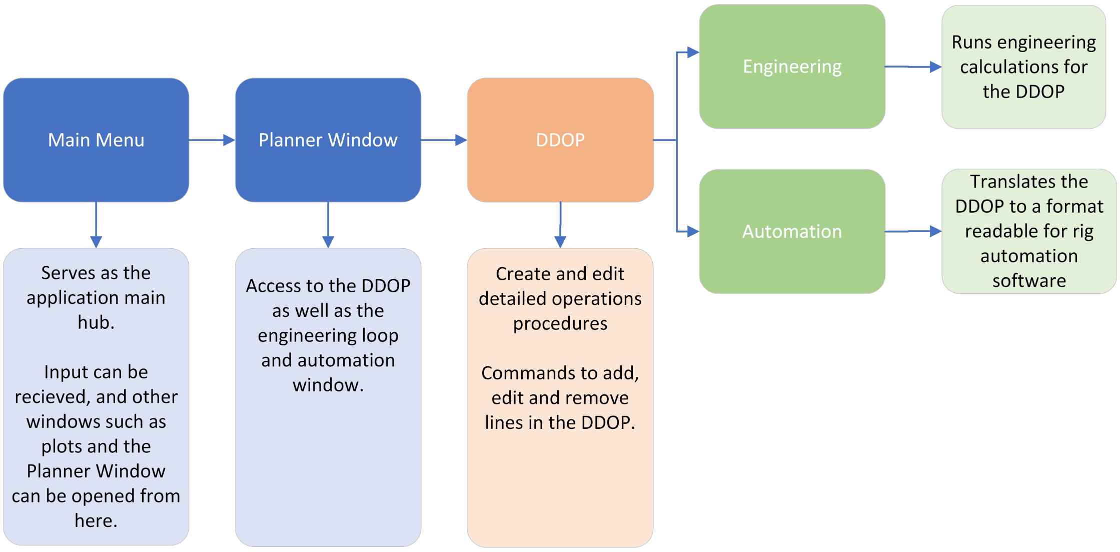


Figure 3.1: Overview of WPS.

3.3 Main menu

The main menu acts as a hub giving access to all other functionality of the application and it is the first window that pops up when running the program. Figure 3.2 shows two early drafts of the main menu design. An important point when the menu was designed, was to make it understandable and navigable for users with little to no prior knowledge about the application. It is also important that the user can input variables in a simple manner here. The main menu allows the user to open new windows such as the plot menu or the planner window, using different buttons. The window was created using the CTk-package as mentioned and illustrated in Section C.7.

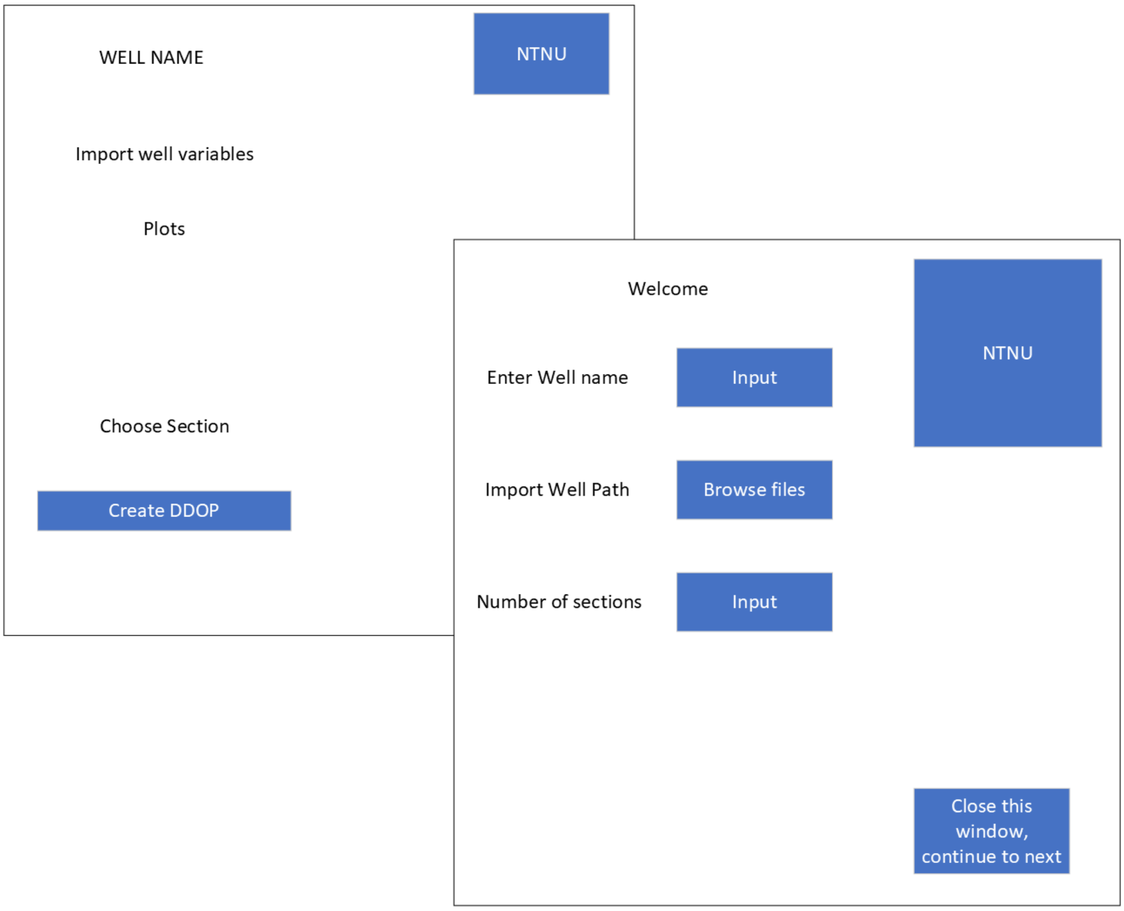


Figure 3.2: Early outline of main menu.

An integral part of making the application work, is that the planner window and other parts of the program must be able to extract information entered in the main menu. For instance, the engineering module must be able to extract variables to perform its calculations. Due to the principle of encapsulation, as per Appendix B, variables can in most cases not be accessed outside its own class. Consequently, the whole application is defined within one class. The input variables are defined as "CTkEntries", allowing them to be accessed in every part of the program using a get-function [1].

---

[1] A get-function returns the value of the key. In this case, the keys are the entry fields where data is inputted.

3.4 Planner window

The planner window is the other main part of the application. Here, the user can create well plans using the DDOP-part of the window. This is also where the user can initiate the engineering module and the automation module that both are connected to the DDOP. All these components are elaborated below.

3.4.1 DDOP

The DDOP-part of the planner window is where actual well plans are created and altered. Figure 3.3 shows an early phase mind-map for the DDOP-part of the application. It covers what this part of the application should contain and touches upon what kind of functionality it should have. Note that this map was made early in the process and does not include all components/functionality that the final DDOP has. Additional information and functionality have been identified later in the development process, such as methods used to make the two modules run.

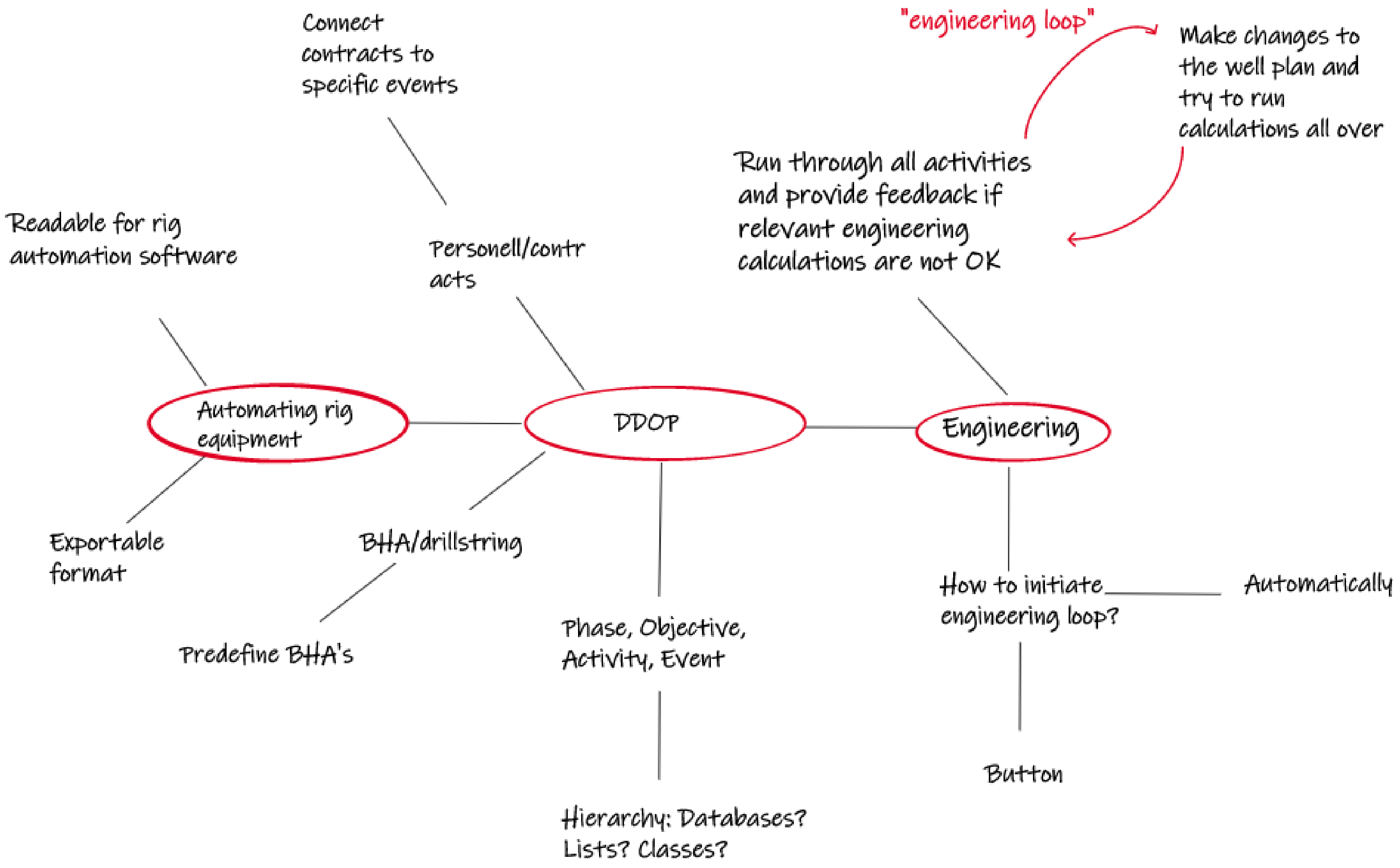


Figure 3.3: Initial mind-map for DDOP.

A key point in the map is ”Phase, Objective, Activity, Event”. This is an important element of the DDOP as it describes how the well operations are defined using a layered information system, see Figure 3.4. The uppermost level, ”Phase”, represents the type of operation in question. This can be drilling, completion, etc. The next level, ”objective” represents different variations of the operation. For drilling this can be ”Drill cement”, ”Drill section” and ”Drill sidetrack”. Layer 3 and 4, ”activity” and ”event” respectively, follow the same procedure. Each layer gives more detailed information.

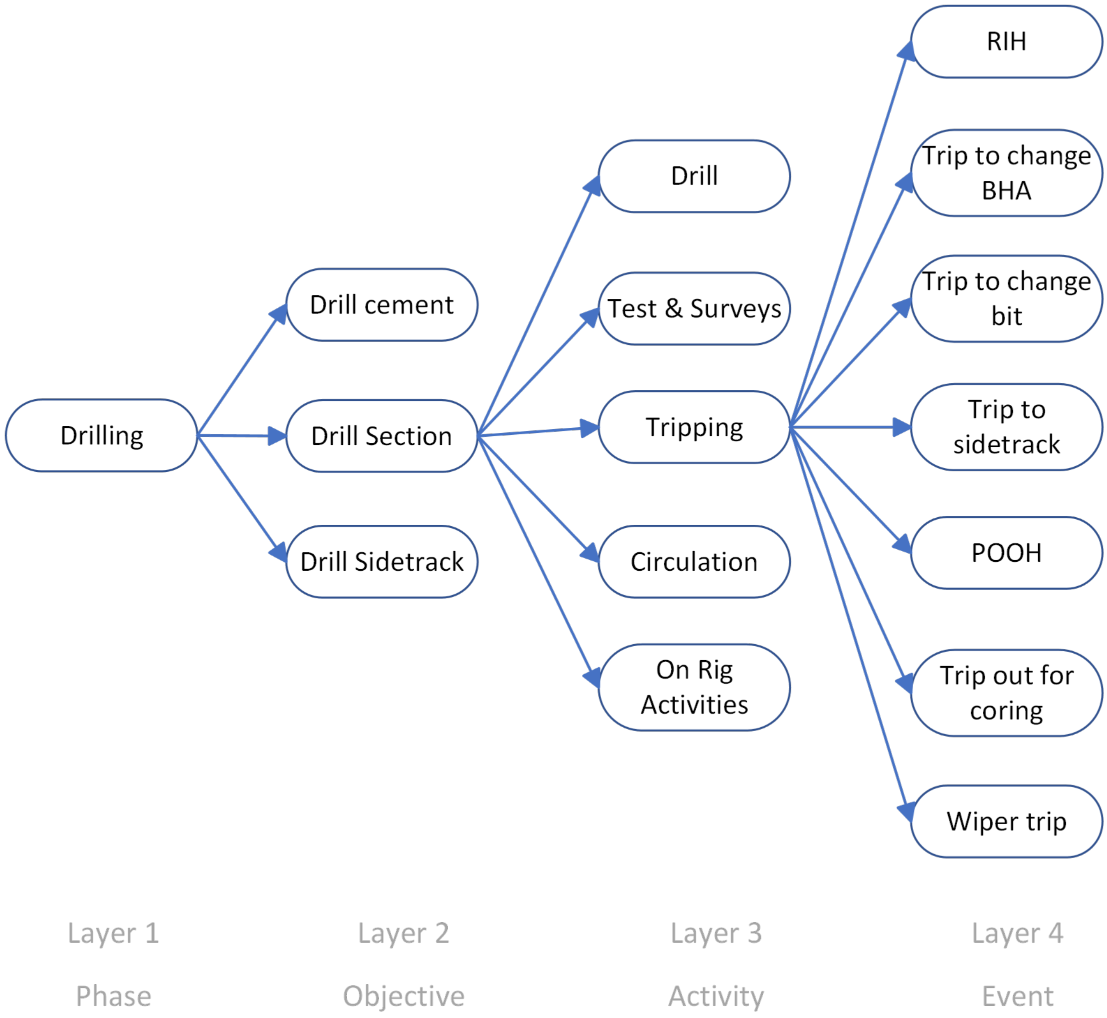


Figure 3.4: Illustration of how information is stored in different layers.

Figure 3.4 also addresses how this hierarchy of information should be defined within the application. This is important as it should be easily accessible when making a well plan. Databases, lists and classes are all mentioned, however, the solution chosen was to define it in a dictionary. This approach uses keys to extract information. For the DDOP, these keys are in fact the same as the different layers. Text strings of phase, objective and activities will give access to information one layer below in the hierarchy. Referring to Figure 3.4, the key ”Drilling” would return values ”Drill cement”, ”Drill section” and ”Drill sidetrack”. Similarly, using two keys in succession will return values in the third layer (activity). Finally, using three keys in succession will return values from the fourth layer (event). This functionality is utilized in the WPS to allow the user to choose which events that should be added to the well plan.

A well plan for a given operation consists of multiple ordered events. In the DDOP, each event is represented with a line. Each line defined is stored in a SQLite3 database. More information on this can be found in Appendix A and Appendix C. In well planning, it is important that the user is in full control. An important design feature is to allow users to change the plans or define parameters they see fit. To create a line in the DDOP, input from the user is required. The user must choose phase, objective, activity and event from drop-down menus, as well as enter start and end depth. Additional special remarks or comments may also be added through an entry field. This is further explained in Section 4.

The DDOP has functionality to let the user add and remove lines. Listing 3.1 shows code that adds a new line to the DDOP, using SQL. Here, input from drop-down boxes and entry field is inserted into the database. Note that for every interaction with the database, one has to connect with the database and create a so-called cursor. Furthermore, commands are used to specify what is to be done with the database. In this case, values should be inserted into the database. The command "INSERT INTO" is therefore used. Other commands are "DROP", "UPDATE" and "DELETE", to name a few.

```python
# Importing sqlite3
import sqlite3

# Defining function to add new line to the DDOP database
def add_record():

        # Connect to the Database
        conn = sqlite3.connect('ddop.db')
        c = conn.cursor()

        # Define placeholders for values that are to be inserted,
        # and adding values that are inserted by user
        c.execute("""INSERT INTO ddop_table VALUES (:phase, :objective, :section,
        :activity, :event, :depth1, :depth2, :contract1, :contract2, :contract3,
        :equipment1, :equipment2, :equipment3,:engineering1, :engineering2,
        :engineering3, :freetext)""",
                {
                        'phase': phase.get(),
                        'objective': objective.get(),
                        'section': section.get(),
                        'activity': activity.get(),
                        'event': event.get(),
                        'depth1': startmd.get(),
                        'depth2': endmd.get(),
                        'contract1': contract1.get(),
                        'contract2': contract2.get(),
                        'contract3': contract3.get(),
                        'equipment1': equipment1.get(),
                        'equipment2': equipment2.get(),
                        'equipment3': equipment3.get(),
                        'engineering1': engineering1.get(),
                        'engineering2': engineering2.get(),
                        'engineering3': engineering3.get(),
                        'freetext': text.get(),
                        'oid': oid.get(),
                })

        conn.commit()
        conn.close()
```

Listing 3.1: Example of adding a line to the DDOP database.

3.4.2 DDOP Engineering module

Another part associated with the DDOP is the engineering module. Here, relevant engineering is run to provide feedback and allow for possible changes to the well plan. The engineering module implemented in the system is per date providing a minimum of data only. As mentioned, not all types of engineering associated with well planning is implemented. However, the engineering module still illustrates how engineering calculations can be implemented in the system and tied directly to the DDOP. Testing this automation concept was prioritized over detailing many calculations. As illustrated in Figure 2.8, the DDOP is a sub-level of the event manager. Figure 3.5 shows how the engineering module is run from the DDOP in the WPS.

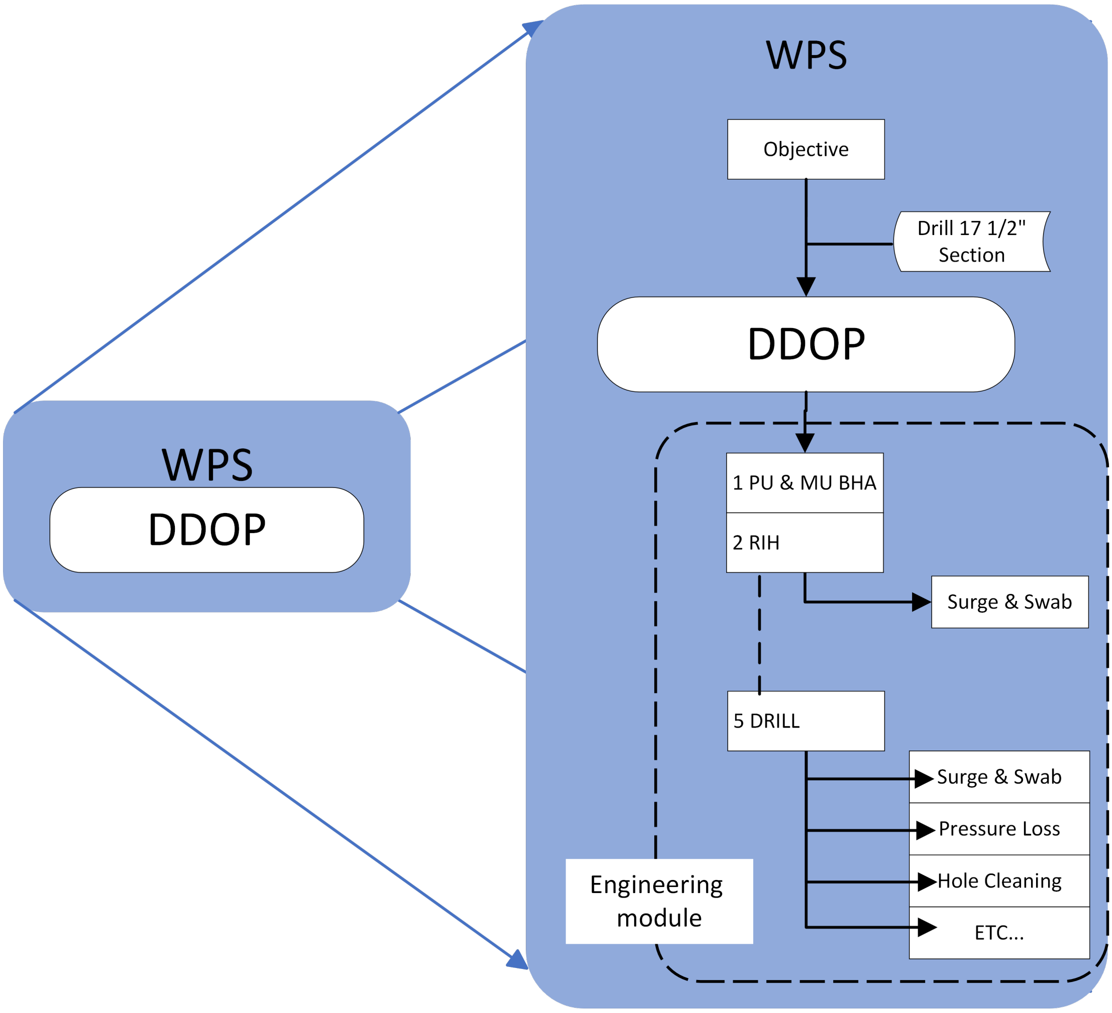


Figure 3.5: Engineering module in WPS.

The engineering loop is ruled by simple logic in Python, see Figure 3.6. It is initiated by the press of the engineering button. The loop runs through every line defined in the DDOP, starting at the very top. For each line, representing one event, it extracts relevant engineering. Every event has pre-defined engineering associated, as exemplified in Table 3.1.

Table 3.1: Example of how well operations are defined with relevant engineering.

Phase	**Objective**	**Activity**	**Event**	**Engineering 1**	**Engineering 2**	**Engineering 3**
Drilling	Drill Section	Initiate drilling	Backream	Surge & Swab	Pressure loss	Hole cleaning

The calculations are run one by one on the line in question. Formulas used were presented in Section 2.4. The calculations are executed and checked against failure criterions established in the code and in Table 3.2. If the criterions are not met, feedback is provided to the user and the opportunity to change relevant parameters is given. This is a simplification of the LCWIM design, where the variable map and control system arrange variables to be within a valid range, or equipment/method is changed. The calculation is then re-run with the updated parameters and checked against the constraints again. If calculations are OK, the loop moves on to the next calculation for the given line. When all relevant engineering for one line has been calculated and is considered OK, the loop moves on to the next line. This process is repeated for every line in the DDOP.

Table 3.2: Engineering failure criterions.

	Failure Criterion
Pressure loss	Pore pressure < Well pressure < Fracturing pressure
Surge & Swab	Pore pressure < Well pressure < Fracturing pressure
Hole cleaning	0.5 < CCI

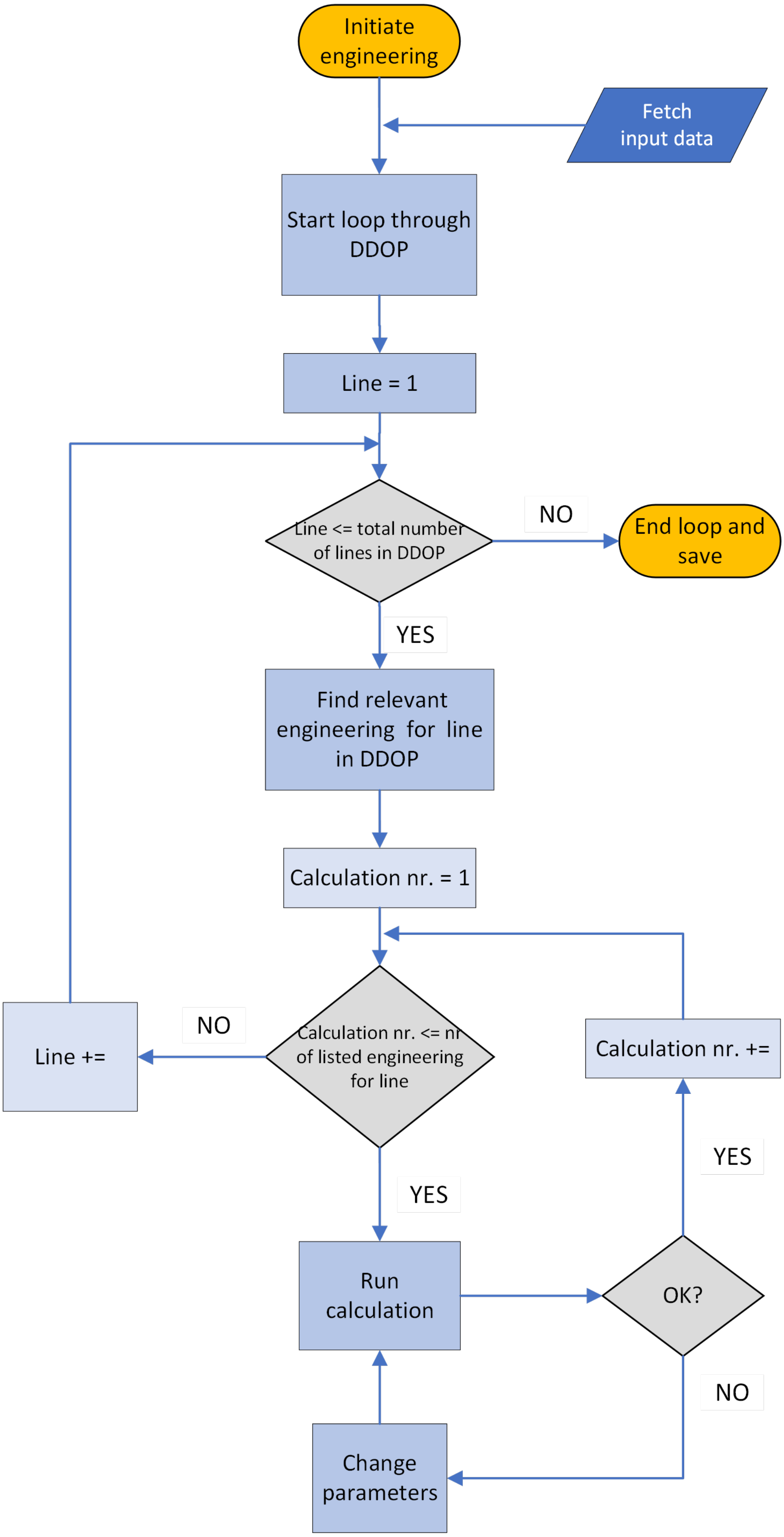


Figure 3.6: Engineering loop.

3.4.3 DDOP Automation module

As pointed out previously, a main objective with this thesis is to make the DDOP in a format readable for humans as well as making it compatible with automated rig equipment and RAS, allowing for automated operations.

To be able to translate the DDOP to a RAS-compatible format, each event must be represented by commands understandable for machines, as mentioned in Section 2.1.5. The approach used in this thesis is to provide instructions to whether equipment is on or off (equipment state), or step up mode for equipment such as pumps, during an event. This means that an event can be represented by equipment being turned on (and start doing what it is intended to do), off (and stop doing what it is doing) or stepped up/down in a specified order. This covers most operations on most rigs. The approach here is to show the functionality. Most equipment is on or off, while other equipment has a step or % range. Many of these can be programmed and predefined, such as NOVOS controls top drive, torque wrench and pipe handling while performing an automated connection. It is a start of full integrity between plan and operation.

Each event in the WPS has been pre-defined with related equipment states. This has been done using a dictionary, as illustrated in Listing 3.2 for "RIH - with BHA on DP". This is a sub event of RIH. Automating operations require very specific instructions and commands. An additional layer of information is therefore implemented here. Due to time limitations this has not been implemented in the normal DDOP yet. The user chooses the correct sub event when using the automation module. "Un rack", "Rack back", "Finger number" etc. are all commands intended to be readable for automation software on drilling rigs. The commands have either a "1" or "0" assigned, indicating the state (on/off) of the equipment during the given operation. The order of commands is important and yet to be fixed. All operations require equipment to stage up in steps, such as start rotation and pumping after connection. This functionality goes hand in hand with NOV's RAS "NOVOS" (NOV, n.d.).

```
dictionary = {
    "RIH":
        {"With BHA on DP":
            {'Un rack':1, 'Rack back':0, 'Finger number':1, 'Finger position':1,
            'Dope':1, 'Change handling equipment':0, 'PU pipe / item':0,
            'LD pipe / item':1, 'Pull slips':1, 'Set slips':1,
            'Set dog collar':0, 'Release dog collar':0, 'Mud bucket on':0,
            'Mud bucket off':0, 'Wiper on':0, 'Wiper off':0, 'DDM position 1':1,
            'DDM position 2':1, 'DDM position 3':1, 'DDM position 4':1,
            'No Rotation':0, 'Rotation 1':1, 'Rotation 2':0, 'Rotation 3':0,
            'Rotation 4':0, 'Stop rotation':1, 'Break circulation':0,
            'Stage up to rate':0, 'Section / High flow':0,
            'Slow (pressure test / fill)':0, 'stop pumping':0, 'no pumping':1,
            'Trip tank open':1, 'Trip tank closed':0, 'Active pit open':0,
            'Active pit closed':1,'Shakers on':0, 'Shakers off':1,
            'Desander on':0, 'Desander off':1,'Desilter on':0,
            'Desilter off':1}
        }
}
```

Listing 3.2: Example of how events are defined in dictionary.

The loop to create the RAS-compatible DDOP in the WPS is shown in Figure 3.7. It is initiated on the press of a button. The program fetches the DDOP database and starts a loop through all events defined in the DDOP. All events are stored in a list as text strings. Also start and end depth for the event in question are stored in other lists. The event list is then run through, with every event-string being looked up in the mentioned dictionary containing equipment states. If found, the user chooses the correct sub event and commands are written to an exportable CSV file. Contrary, if the string is not found, the process moves on to the next event. This suggest that the nature of this specific event is that of not needing to be automated, e.g. pre-job meeting. This is repeated until the very last line. Ultimately, the CSV file will contain equipment states to all events in the DDOP.

In execution, the DDOP is then a guide for both the driller and the RAS, giving the operation another layer of safety. The RAS follows company procedures for every single event, and the driller confirms when the event is complete and the next is okay to start. The DDOP with the full overview of state of the equipment provides a more efficient and safe operation. One perspective is the correct actions, another is to verify "operational health". Replacing planned parameters with the real in addition to knowing the exact state and phase of the operation, enables detection of downhole problems and well control issues and probably the quickest path to correct mitigating and corrective actions.

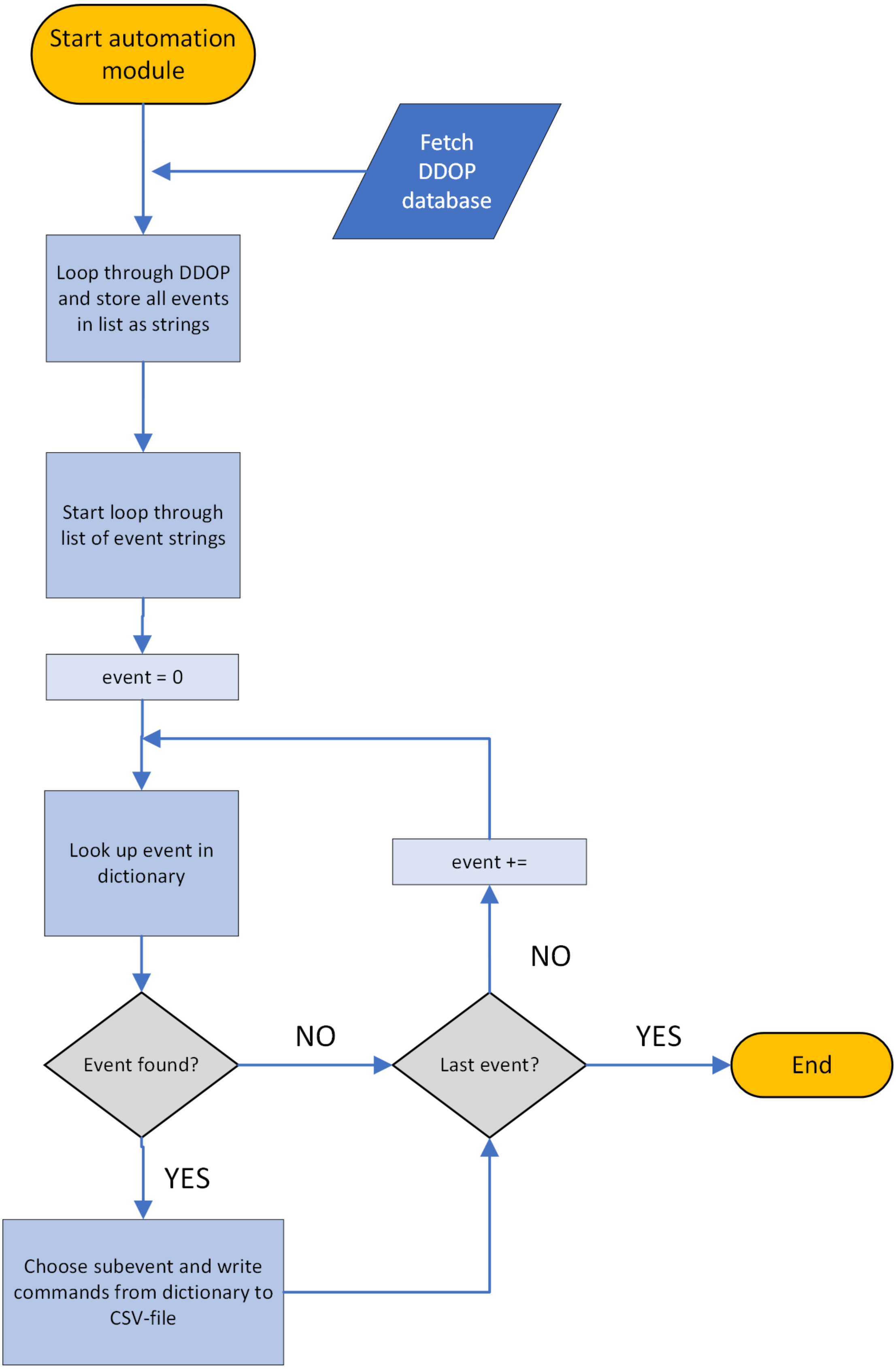


Figure 3.7: Simplified automation module flow chart.

3.5 Standardized equipment and contracts

Many elements of the prototype have been standardized. This is done to streamline the work flow and avoid repetitive work. In the WPS, two BHAs have been pre-defined, see Table 3.3 and Table 3.4. This is done since the same type of BHA is often used in wells with similar characteristics. The user can change BHA if calculations fail in the engineering loop.

When choosing a standardized BHA, one must consider the flow rate needed to have sufficient hole cleaning without exceeding the pressure limits in the well. Additionally, tools such as measurement while drilling (MWD) and the mud motor are dependent on the flow rate. BHA # 1 & #2 can be detailed with a high flow and a medium flow version.

In a 17 1/2" section, as discussed in this thesis, the equivalent circulating density (ECD) varies with 2-3 points for the most part. This is also true for deviated holes that require higher flow rates. In the generic procedure used in this thesis, the pump rate is restricted by size of pipe (ID) and other BHA properties. This thesis did not detail the BHA design further to focus on testing the concept of connecting procedure, engineering and tools.

Furthermore, each event have been pre-defined with contract names. These can be used as keys to unlock full contracts at a later stage. This approach is adequate as the same operations usually call for the same types of contracts. For some cases there might be a need to change the information that already has been defined, or even add more information. This can be done by editing the lines in the DDOP, or adding information to the description box of the DDOP.

Table 3.3: BHA #1 implemented in the WPS.

BHA item	Length (m)	OD (in)	ID (in)
Bit	0.43	17 1/2	
Motor	8.85	11 1/4	9 1/4
Sub	0.80	9 1/2	3
Stab.	2.00	9 1/2	3
Logging tool	6.80	9 1/2	4 17/20
MWD	7.60	9 1/2	6 1/4
Stab	2.00	9 1/2	3
NMDC	18.80	9 1/2	3
Crossover	0.30	9	2 5/8
DC	55.80	8	2 7/8
Jar	9.60	8	3
DC	18.60	8	2 7/8
Crossover	0.50	5 1/2	2 1/2
HWDP	138.00	5 1/2	3 1/4
DP	To surface	5 1/2	4 7/9

Table 3.4: BHA #2 implemented in the WPS.

BHA item	Length (m)	OD (in)	ID (in)
Rock Bit	0.30	17 1/2	
Motor	10.40	12 3/4	3
Stab.	2.00	9 1/2	3
Sub	0.80	9 1/2	3
MWD	10.50	9 1/2	3
Sub	0.80	9 1/2	3
Stab.	2.00	8	3
DC	9.50	8	3
Stab.	2.00	8	3
DC	9.50	8	3
Jar	9.50	8	3
DC	28.50	8	3
Sub	1.00	8	3
HWDP	112.00	6 5/8	4 46/63
DP	To surface	5 1/2	4 7/9

(Intentionally left blank.)

4 Results

The WPS is a prototype application developed in python, to perform automated well planning in accordance with the software framework outlined by Brechan (2020). The WPS was developed to provide a digital program/DDOP in formats readable for humans and RAS, and show how engineering, equipment and contracts can be integrated with this.

4.1 Main menu

Figure 4.1 shows the main menu of the WPS. The main menu is split into two different frames, one on the left hand side and one on the right hand side. In the left frame general input for the well can be inputted, such as the name of the well, number of sections and the active section currently being worked on. It is also possible to import the wellpath of the well. In the right frame of the main menu, input being used in the engineering loop can be entered. These are variables are connected to the implemented hydraulic calculations of the WPS. From the right frame the plot menu can be opened if the "Plots" button is pressed, and the planner window can be opened by pressing the "Open Planner" button.

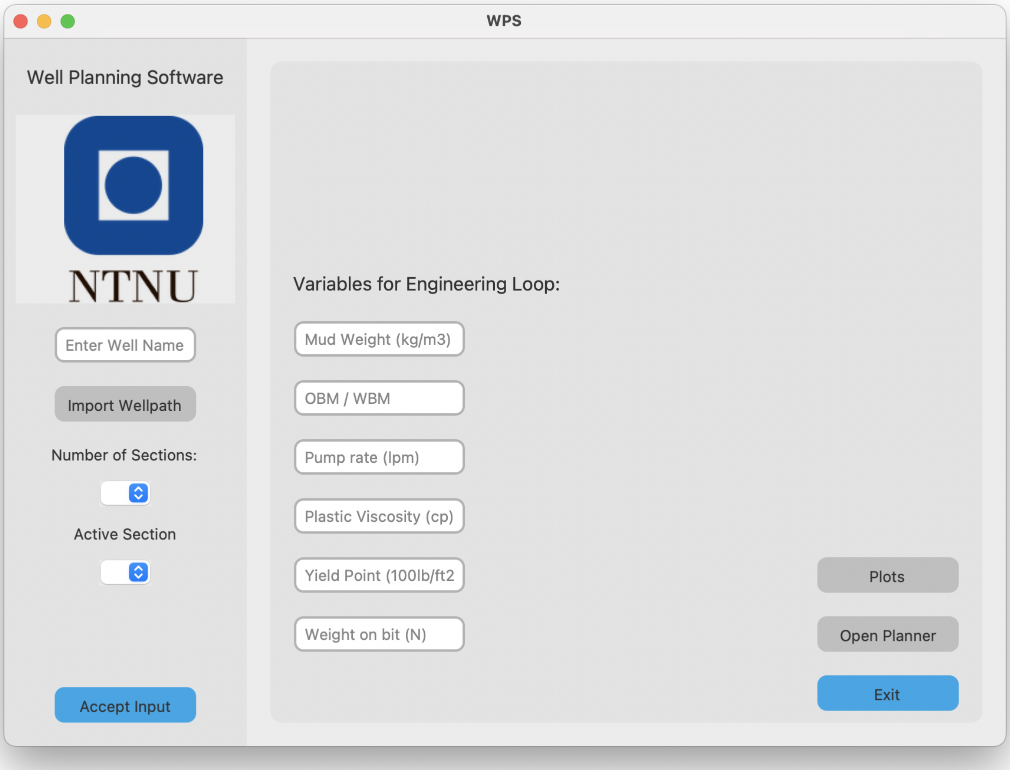
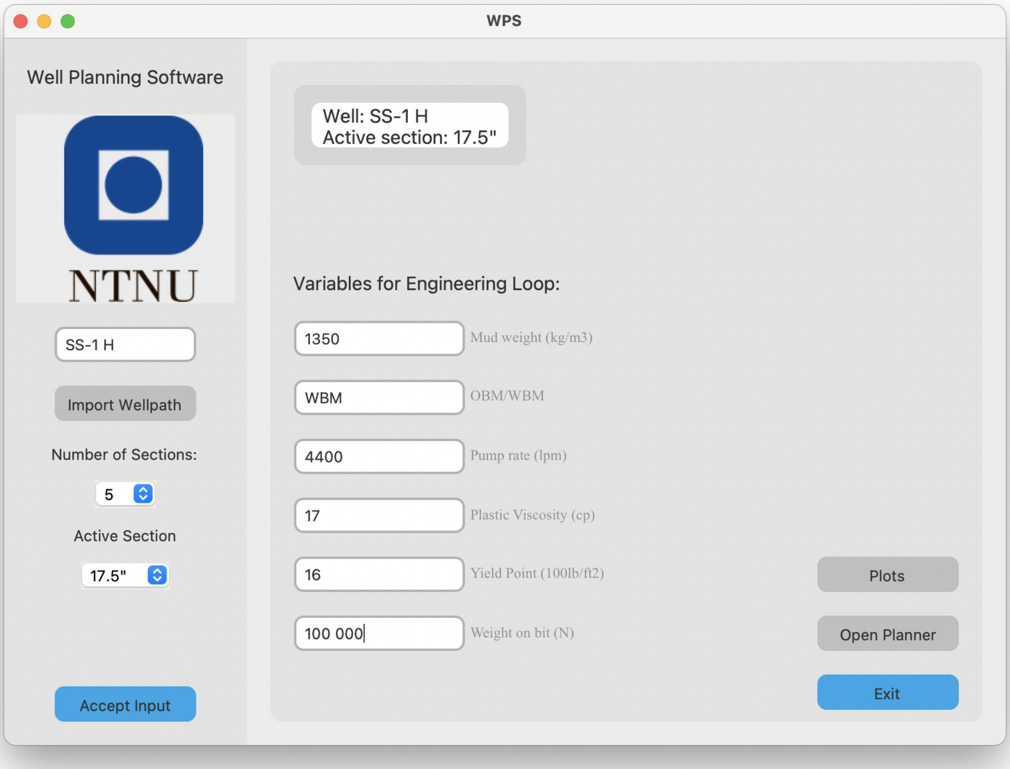


Figure 4.1: Main menu.

Figure 4.2 shows the main menu of the WPS with input in both frames. The name of the well, SS-1 H has been entered, a wellpath has been selected and the two drop-down menus are both in use. Once the "Accept Input" button has been pressed, the name of the well as well as the active section will appear in the right frame of the main menu inside a message box. This gives the user confirmation that the input has been entered successfully. Beneath the message box, variables for the engineering loop has been entered. The values that is used in the figure is given in the DOP that has been digitalized, and can be seen in Appendix E. Once all input has been entered the user can continue to either the plot menu or the planner window.



SS-1 H

Number of Sections: 5

Active Section 17.5"

Well: SS-1 H
Active section: 17.5"

1350 Mud weight (kg/m3)

WBM OBM/WBM

4400 Pump rate (lpm)

17 Plastic Viscosity (cp)

16 Yield Point (100lb/ft2)

100 000 Weight on bit (N)

Figure 4.2: Main menu with input.

4.2 Plot menu

The plot menu produces four buttons at this stage of the development. One button for each plot included in the application, and one for returning to the main menu. The plots produce a visual representation of the imported wellpath from different perspectives. Once a button has been clicked a new window will open with the corresponding plot. The plot menu and the plots are given in Figure 4.3.

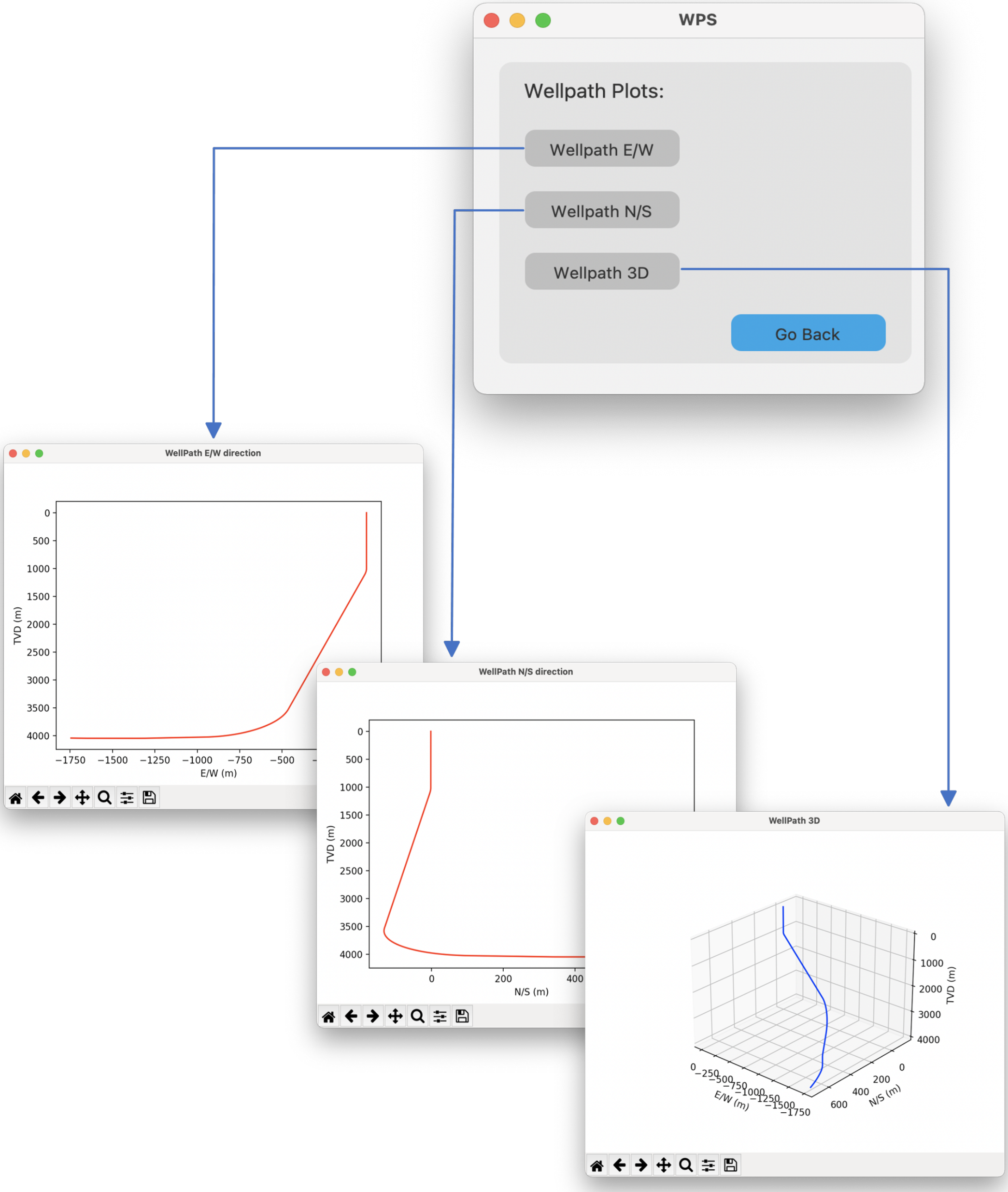


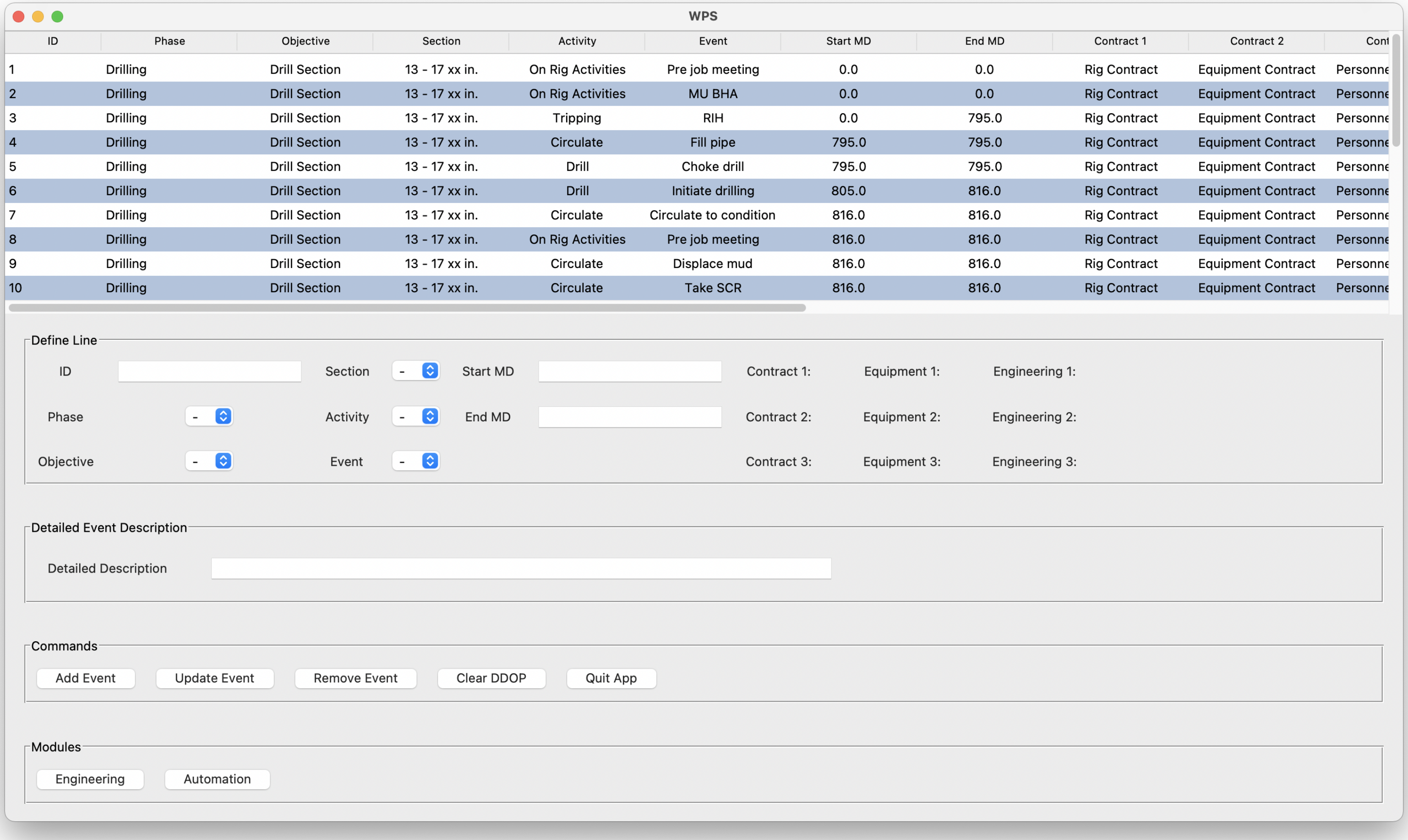
Figure 4.3: Plot menu with plots: (TVD v E/W), (TVD v N/S) and (TVD v E/W v N/S).

4.3 Planner window

Figure 4.4 shows the planner window, which is the main part of the WPS. The window is divided into five different frames from top to bottom. The first frame shows a generic DDOP. This frame is referred to as the DDOP-frame and is where the database containing the DDOP is illustrated and shown to the user. The second frame is called the "Define Line"-frame and is used for defining the events that should be inserted into the DDOP. This section can also be used to modify already existing events. The next frame, called the "Detailed Event Description" frame, contains an input box where the user can add information in form of free text to each event of the DDOP. The fourth frame of the planner window contains the commands to edit the DDOP. The information inputted in the two previous frames are first fetched once a button is pressed in this frame. There are in total five buttons with different functionality:

- "Add Event": adds a new line to the DDOP.
- "Update Event" edits an already existing line.
- "Remove Event" removes a line from the DDOP.
- "Clear DDOP": deletes all lines in the DDOP.
- "Quit App": exits the application.

The last frame called "Modules" contains two buttons. The first being the "Engineering" button, and the second being the "Automation" button. These buttons initiate the engineering loop and the automation loop as described in Section 3.4.2 and Section 3.4.3, respectively.



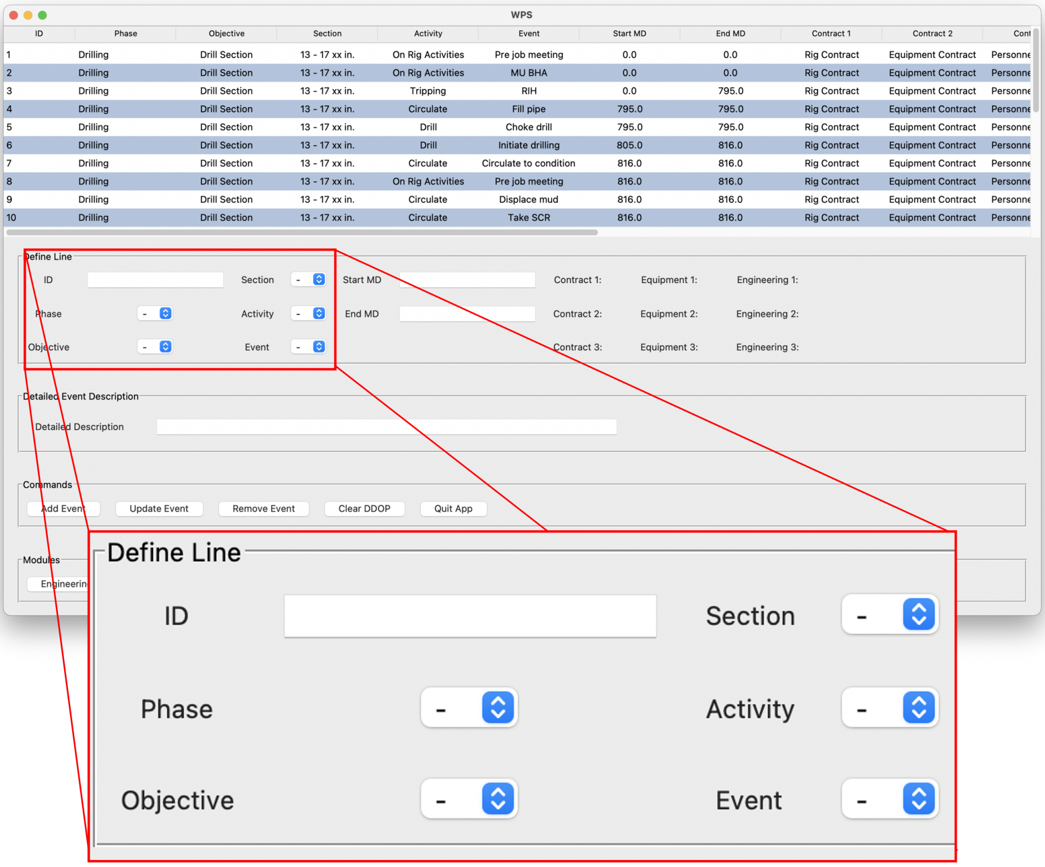
ID	Phase	Objective	Section	Activity	Event	Start MD	End MD	Contract 1	Contract 2	Cont
1	Drilling	Drill Section	13 - 17 xx in.	On Rig Activities	Pre job meeting	0.0	0.0	Rig Contract	Equipment Contract	Personne
2	Drilling	Drill Section	13 - 17 xx in.	On Rig Activities	MU BHA	0.0	0.0	Rig Contract	Equipment Contract	Personne
3	Drilling	Drill Section	13 - 17 xx in.	Tripping	RIH	0.0	795.0	Rig Contract	Equipment Contract	Personne
4	Drilling	Drill Section	13 - 17 xx in.	Circulate	Fill pipe	795.0	795.0	Rig Contract	Equipment Contract	Personne
5	Drilling	Drill Section	13 - 17 xx in.	Drill	Choke drill	795.0	795.0	Rig Contract	Equipment Contract	Personne
6	Drilling	Drill Section	13 - 17 xx in.	Drill	Initiate drilling	805.0	816.0	Rig Contract	Equipment Contract	Personne
7	Drilling	Drill Section	13 - 17 xx in.	Circulate	Circulate to condition	816.0	816.0	Rig Contract	Equipment Contract	Personne
8	Drilling	Drill Section	13 - 17 xx in.	On Rig Activities	Pre job meeting	816.0	816.0	Rig Contract	Equipment Contract	Personne
9	Drilling	Drill Section	13 - 17 xx in.	Circulate	Displace mud	816.0	816.0	Rig Contract	Equipment Contract	Personne
10	Drilling	Drill Section	13 - 17 xx in.	Circulate	Take SCR	816.0	816.0	Rig Contract	Equipment Contract	Personne

Figure 4.4: The planner window populated with events.

4.4 DDOP

4.4.1 Adding new lines to DDOP

When adding a new line (event) to the DDOP, the user has to choose different options from the drop-down menus. These are shown in Figure 4.5.

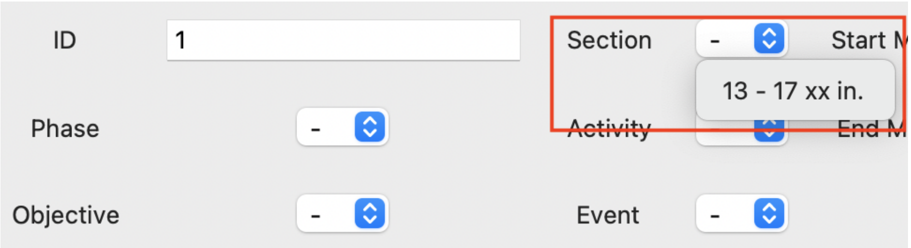


ID	Phase	Objective	Section	Activity	Event	Start MD	End MD	Contract 1	Contract 2	Cont
1	Drilling	Drill Section	13 - 17 xx in.	On Rig Activities	Pre job meeting	0.0	0.0	Rig Contract	Equipment Contract	Personne
2	Drilling	Drill Section	13 - 17 xx in.	On Rig Activities	MU BHA	0.0	0.0	Rig Contract	Equipment Contract	Personne
3	Drilling	Drill Section	13 - 17 xx in.	Tripping	RIH	0.0	795.0	Rig Contract	Equipment Contract	Personne
4	Drilling	Drill Section	13 - 17 xx in.	Circulate	Fill pipe	795.0	795.0	Rig Contract	Equipment Contract	Personne
5	Drilling	Drill Section	13 - 17 xx in.	Drill	Choke drill	795.0	795.0	Rig Contract	Equipment Contract	Personne
6	Drilling	Drill Section	13 - 17 xx in.	Drill	Initiate drilling	805.0	816.0	Rig Contract	Equipment Contract	Personne
7	Drilling	Drill Section	13 - 17 xx in.	Circulate	Circulate to condition	816.0	816.0	Rig Contract	Equipment Contract	Personne
8	Drilling	Drill Section	13 - 17 xx in.	On Rig Activities	Pre job meeting	816.0	816.0	Rig Contract	Equipment Contract	Personne
9	Drilling	Drill Section	13 - 17 xx in.	Circulate	Displace mud	816.0	816.0	Rig Contract	Equipment Contract	Personne
10	Drilling	Drill Section	13 - 17 xx in.	Circulate	Take SCR	816.0	816.0	Rig Contract	Equipment Contract	Personne

Section -
Phase -
Activity -
Objective -
Event -

Figure 4.5: Drop-down menus for adding new line to DDOP.

First, the user selects the section in question, as shown in Figure 4.6.



ID 1
Section 13 - 17 xx in.
Phase -
Objective -
Event -

Figure 4.6: Adding new event to DDOP (1/6).

The user then chooses the correct phase for the event, as shown Figure 4.7. Only drilling is implemented in the system per date.

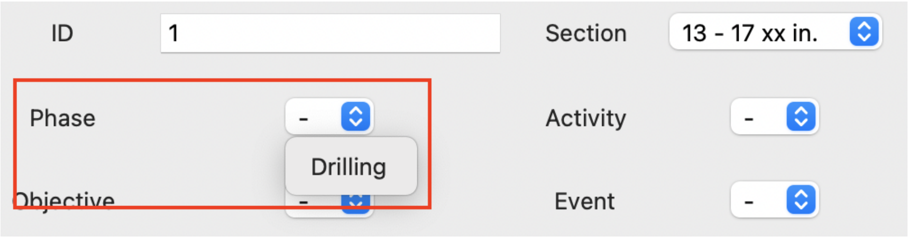

ID 1
Section 13 - 17 xx in.
Phase -
Activity -
Objective -
Event -

Figure 4.7: Adding new event to DDOP (2/6).

The drop-down menus are dependent on each other and will be updated based on whats inputted earlier, as per Section 3.4.1. After phase is selected, the user picks the objective, as Figure 4.8 shows.

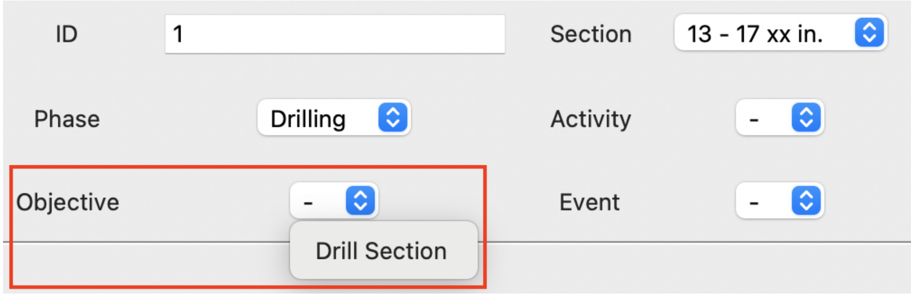

ID 1
Section 13 - 17 xx in.
Phase Drilling
Activity -
Objective -
Event -

Figure 4.8: Adding new event to DDOP (3/6).

Different activities can be performed within the objective "Drill Section", as shown in Figure 4.9. The user chooses the correct one.

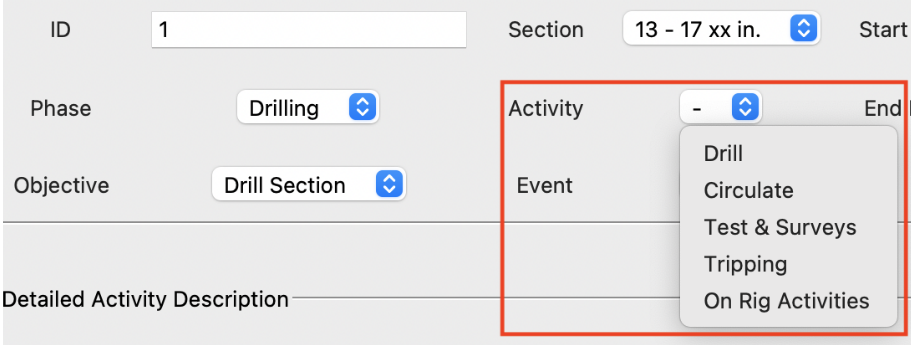
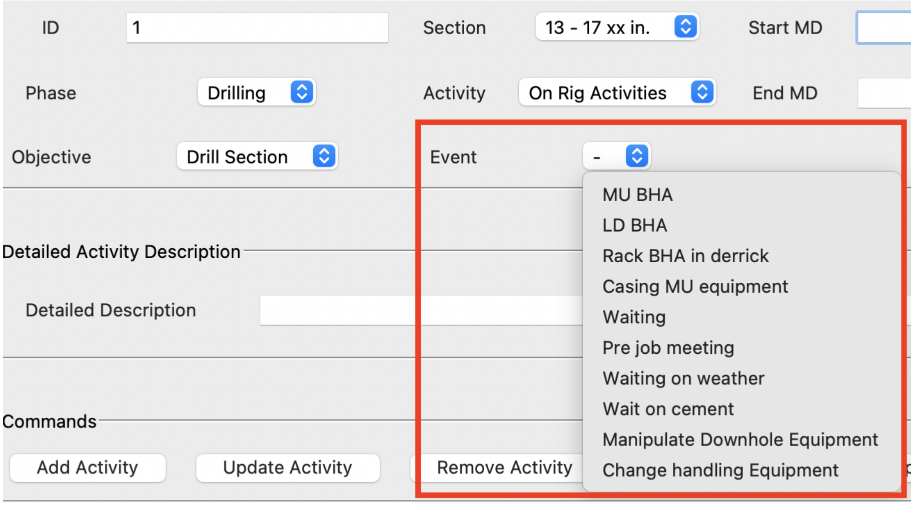

ID 1
Section 13 - 17 xx in.
Start
Phase Drilling
Activity -
End
Objective Drill Section
Event
Detailed Activity Description

Figure 4.9: Adding new event to DDOP (4/6).

Lastly, the user chooses the event that should be inserted in the DDOP. Alternatives given in drop-down menu are based on previous choices. See Figure 4.10.



ID 1

Section 13 - 17 xx in.

Start MD

Phase Drilling

Activity On Rig Activities

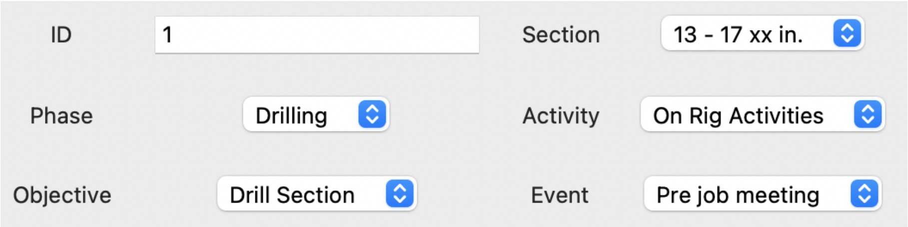
End MD

Objective Drill Section

Event -

Figure 4.10: Adding new event to DDOP (5/6).

Figure 4.11 shows what is inserted in the DDOP. Additionally, the user inputs start and end depth for the event in question.



ID 1

Section 13 - 17 xx in.

Phase Drilling

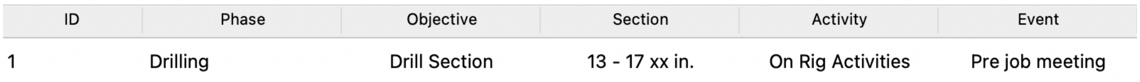
Activity On Rig Activities

Objective Drill Section

Event Pre job meeting

Figure 4.11: Adding new event to DDOP (6/6).

Event inserted in the DDOP is shown in Figure 4.12. If the user wants to edit an already existing event, the user clicks on the line with the relevant event, which automatically updates the drop-down menus and information boxes with the information already defined, and change the event using the same procedure.

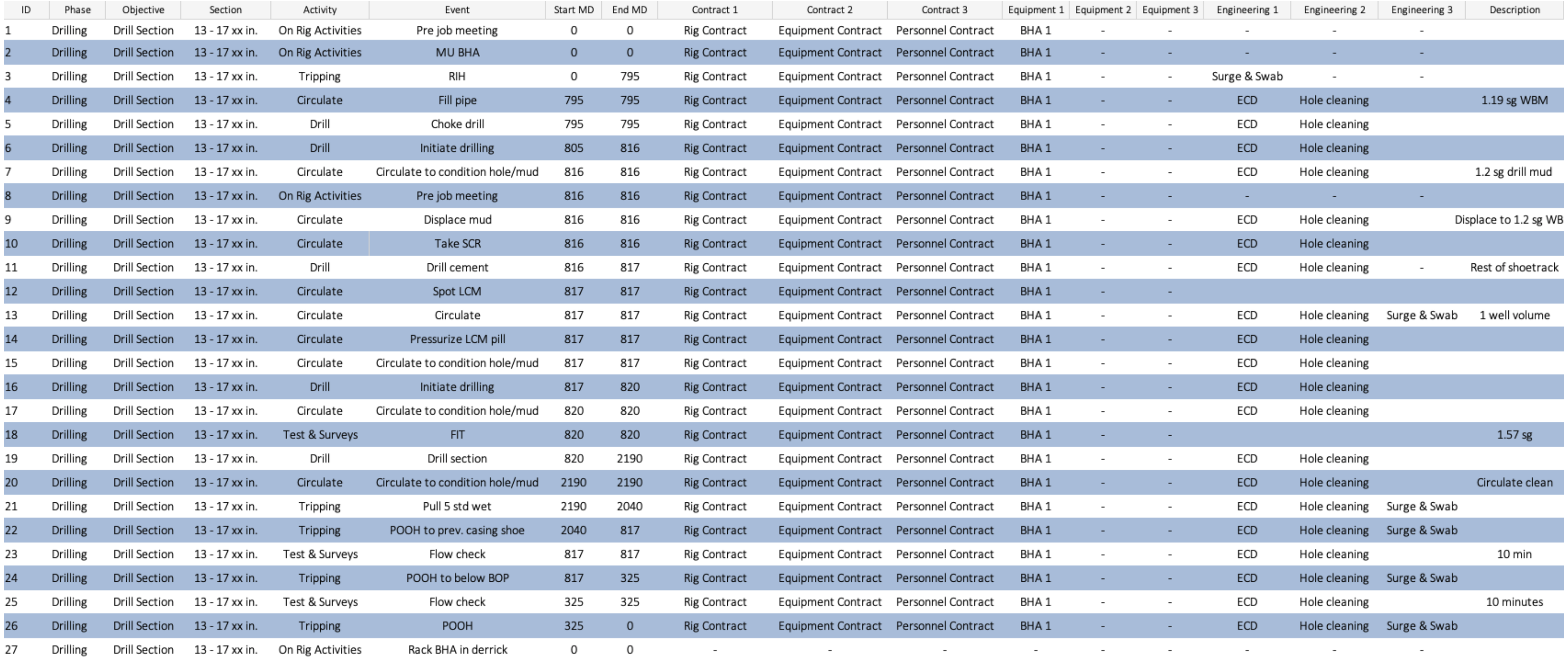


ID	Phase	Objective	Section	Activity	Event
1	Drilling	Drill Section	13 - 17 xx in.	On Rig Activities	Pre job meeting

Figure 4.12: Event inserted in DDOP.

4.4.2 Standard DDOP

Figure 4.13 shows a standard DDOP produced by adding a number of events in a given order. As can be seen, each event is defined with start- and end depth. This interval indicates the planned measured depth at which the event will take place. Further, all events, where relevant, are defined with standard contracts, equipment and engineering. Some events also have specific remarks, such as mud weight or other instructions.



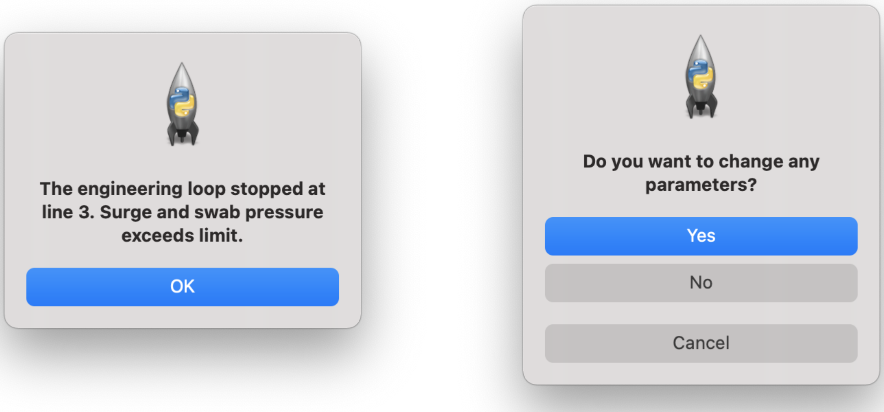
ID	Phase	Objective	Section	Activity	Event	Start MD	End MD	Contract 1	Contract 2	Contract 3	Equipment 1	Equipment 2	Equipment 3	Engineering 1	Engineering 2	Engineering 3	Description
1	Drilling	Drill Section	13 - 17 xx in.	On Rig Activities	Pre job meeting	0	0	Rig Contract	Equipment Contract	Personnel Contract	BHA 1	-	-	-	-	-	
2	Drilling	Drill Section	13 - 17 xx in.	On Rig Activities	MU BHA	0	0	Rig Contract	Equipment Contract	Personnel Contract	BHA 1	-	-	-	-	-	
3	Drilling	Drill Section	13 - 17 xx in.	Tripping	RIH	0	795	Rig Contract	Equipment Contract	Personnel Contract	BHA 1	-	-	Surge & Swab	-	-	
4	Drilling	Drill Section	13 - 17 xx in.	Circulate	Fill pipe	795	795	Rig Contract	Equipment Contract	Personnel Contract	BHA 1	-	-	ECD	Hole cleaning		1.19 sg WBM
5	Drilling	Drill Section	13 - 17 xx in.	Drill	Choke drill	795	795	Rig Contract	Equipment Contract	Personnel Contract	BHA 1	-	-	ECD	Hole cleaning		
6	Drilling	Drill Section	13 - 17 xx in.	Drill	Initiate drilling	805	816	Rig Contract	Equipment Contract	Personnel Contract	BHA 1	-	-	ECD	Hole cleaning		
7	Drilling	Drill Section	13 - 17 xx in.	Circulate	Circulate to condition hole/mud	816	816	Rig Contract	Equipment Contract	Personnel Contract	BHA 1	-	-	ECD	Hole cleaning		1.2 sg drill mud
8	Drilling	Drill Section	13 - 17 xx in.	On Rig Activities	Pre job meeting	816	816	Rig Contract	Equipment Contract	Personnel Contract	BHA 1	-	-	-	-	-	
9	Drilling	Drill Section	13 - 17 xx in.	Circulate	Displace mud	816	816	Rig Contract	Equipment Contract	Personnel Contract	BHA 1	-	-	ECD	Hole cleaning		Displace to 1.2 sg WB
10	Drilling	Drill Section	13 - 17 xx in.	Circulate	Take SCR	816	816	Rig Contract	Equipment Contract	Personnel Contract	BHA 1	-	-	ECD	Hole cleaning		
11	Drilling	Drill Section	13 - 17 xx in.	Drill	Drill cement	816	817	Rig Contract	Equipment Contract	Personnel Contract	BHA 1	-	-	ECD	Hole cleaning	-	Rest of shoetrack
12	Drilling	Drill Section	13 - 17 xx in.	Circulate	Spot LCM	817	817	Rig Contract	Equipment Contract	Personnel Contract	BHA 1	-	-				
13	Drilling	Drill Section	13 - 17 xx in.	Circulate	Circulate	817	817	Rig Contract	Equipment Contract	Personnel Contract	BHA 1	-	-	ECD	Hole cleaning	Surge & Swab	1 well volume
14	Drilling	Drill Section	13 - 17 xx in.	Circulate	Pressurize LCM pill	817	817	Rig Contract	Equipment Contract	Personnel Contract	BHA 1	-	-	ECD	Hole cleaning		
15	Drilling	Drill Section	13 - 17 xx in.	Circulate	Circulate to condition hole/mud	817	817	Rig Contract	Equipment Contract	Personnel Contract	BHA 1	-	-	ECD	Hole cleaning		
16	Drilling	Drill Section	13 - 17 xx in.	Drill	Initiate drilling	817	820	Rig Contract	Equipment Contract	Personnel Contract	BHA 1	-	-	ECD	Hole cleaning		
17	Drilling	Drill Section	13 - 17 xx in.	Circulate	Circulate to condition hole/mud	820	820	Rig Contract	Equipment Contract	Personnel Contract	BHA 1	-	-	ECD	Hole cleaning		
18	Drilling	Drill Section	13 - 17 xx in.	Test & Surveys	FIT	820	820	Rig Contract	Equipment Contract	Personnel Contract	BHA 1	-	-				1.57 sg
19	Drilling	Drill Section	13 - 17 xx in.	Drill	Drill section	820	2190	Rig Contract	Equipment Contract	Personnel Contract	BHA 1	-	-	ECD	Hole cleaning		
20	Drilling	Drill Section	13 - 17 xx in.	Circulate	Circulate to condition hole/mud	2190	2190	Rig Contract	Equipment Contract	Personnel Contract	BHA 1	-	-	ECD	Hole cleaning		Circulate clean
21	Drilling	Drill Section	13 - 17 xx in.	Tripping	Pull 5 std wet	2190	2040	Rig Contract	Equipment Contract	Personnel Contract	BHA 1	-	-	ECD	Hole cleaning	Surge & Swab	
22	Drilling	Drill Section	13 - 17 xx in.	Tripping	POOH to prev. casing shoe	2040	817	Rig Contract	Equipment Contract	Personnel Contract	BHA 1	-	-	ECD	Hole cleaning	Surge & Swab	
23	Drilling	Drill Section	13 - 17 xx in.	Test & Surveys	Flow check	817	817	Rig Contract	Equipment Contract	Personnel Contract	BHA 1	-	-	ECD	Hole cleaning		10 min
24	Drilling	Drill Section	13 - 17 xx in.	Tripping	POOH to below BOP	817	325	Rig Contract	Equipment Contract	Personnel Contract	BHA 1	-	-	ECD	Hole cleaning	Surge & Swab	
25	Drilling	Drill Section	13 - 17 xx in.	Test & Surveys	Flow check	325	325	Rig Contract	Equipment Contract	Personnel Contract	BHA 1	-	-	ECD	Hole cleaning		10 minutes
26	Drilling	Drill Section	13 - 17 xx in.	Tripping	POOH	325	0	Rig Contract	Equipment Contract	Personnel Contract	BHA 1	-	-	ECD	Hole cleaning	Surge & Swab	
27	Drilling	Drill Section	13 - 17 xx in.	On Rig Activities	Rack BHA in derrick	0	0	-	-	-	-	-	-	-	-	-	

Figure 4.13: DDOP exclusive rig equipment state.

4.5 DDOP Engineering module

The engineering loop is initiated by pressing the "Engineering"-button. The DDOP is fetched from the database and the loop starts. If any calculations fail, the user is notified and may try to fix the problem. This is exemplified in Figure 4.14 and Figure 4.15, where calculations have failed due to surge & swab effects and due to ECD, respectively.

Note that the user is given the option to changer either mud weight or BHA. More options could be implemented later. If the user wants to change BHA, a different pre-defined BHA is used as per Section 3.5.

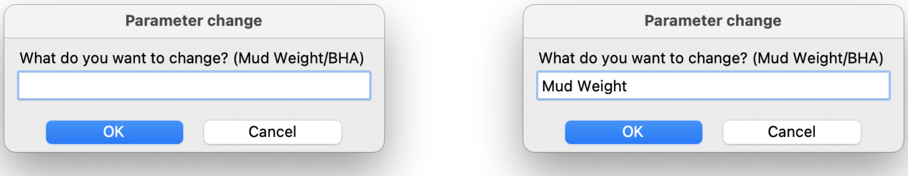


The engineering loop stopped at line 3. Surge and swab pressure exceeds limit.

Do you want to change any parameters?

(a) Loop stopped at line 3 due to surge & swab pressure exceeding limit.

(b) User given option to change any parameters.



Parameter change

What do you want to change? (Mud Weight/BHA)

Parameter change

What do you want to change? (Mud Weight/BHA)

Mud Weight

(c) User enters what should be changed.

(d) User want to change mud weight.

Figure 4.14: Engineering loop feedback example 1.

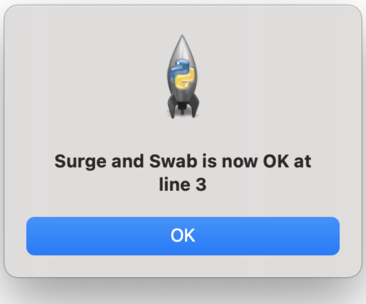
The current mud weight is 1350. Enter new mud weight (kg/m3)!

The current mud weight is 1350. Enter new mud weight (kg/m3)!

1200

(e) Current mud weight given.

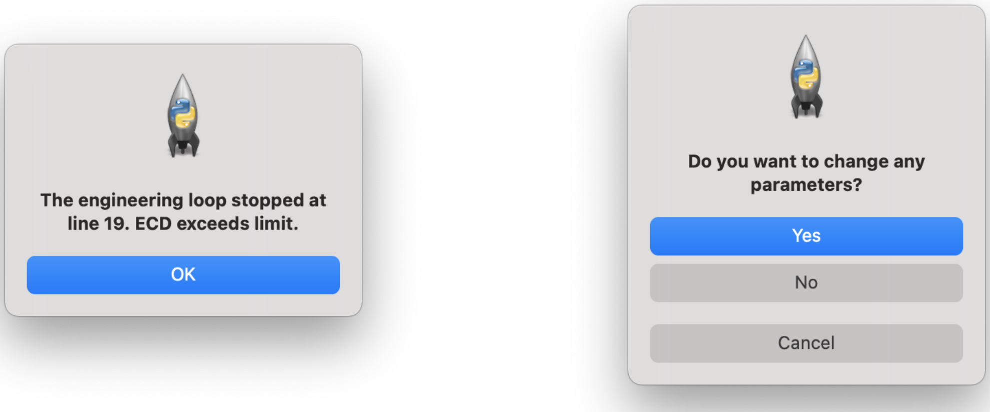
(f) New mud weight entered.



Surge and Swab is now OK at line 3

(g) Feedback to calculations run with updated mud weight.

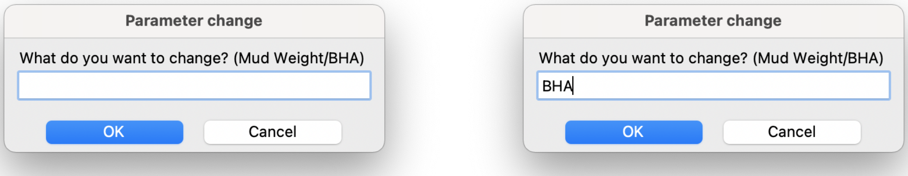
Figure 4.14: Engineering loop feedback example 1 (cont.).


The engineering loop stopped at line 19. ECD exceeds limit.

Do you want to change any parameters?

(a) Loop stopped at line 19 due to ECD pressure exceeding limit.

(b) User given option to change any parameters.


Parameter change

What do you want to change? (Mud Weight/BHA)

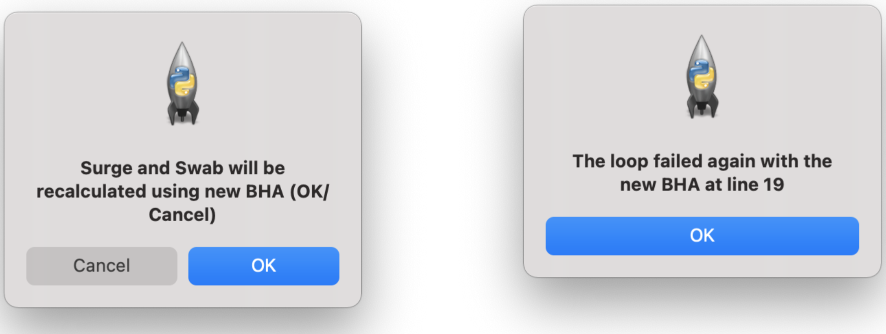
Parameter change

What do you want to change? (Mud Weight/BHA)

BHA

(c) Parameter change options.

(d) User chooses what should be changed.

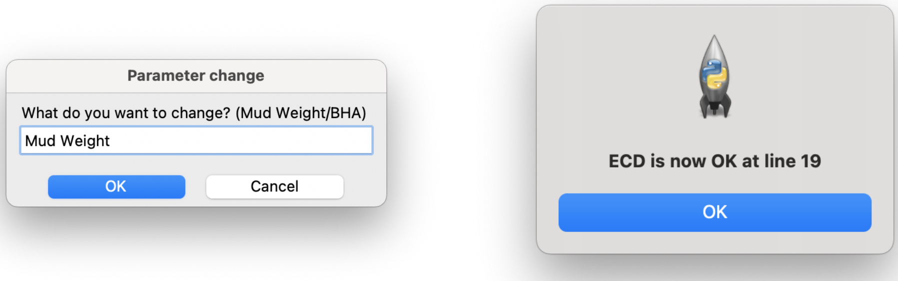

Surge and Swab will be recalculated using new BHA (OK/ Cancel)

The loop failed again with the new BHA at line 19

(e) Calculations are re-run with new BHA.

(f) Calculations did not meet criterions.

Figure 4.15: Engineering loop feedback example 2.



Parameter change

What do you want to change? (Mud Weight/BHA)

Mud Weight

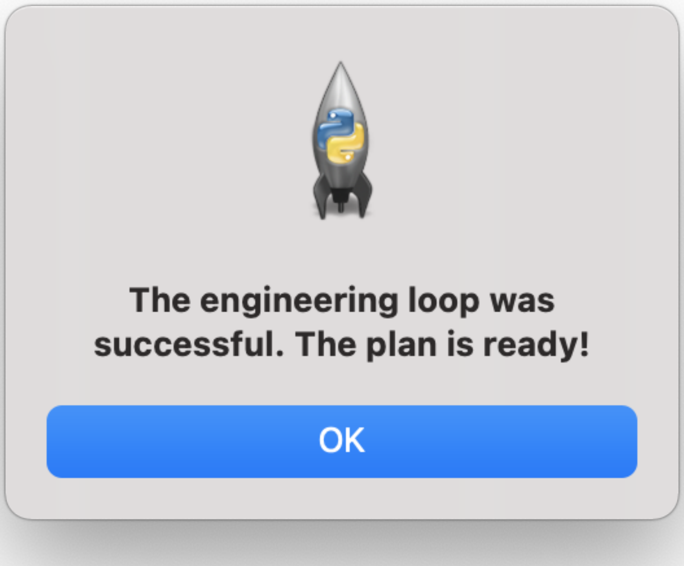
ECD is now OK at line 19

(g) Option to change parameters given again. Mud weight changed as shown in Figure 4.14.

(h) Calculations run with updated mud weight. Criterions are met.

Figure 4.15: Engineering loop feedback example 2 (cont.).

When the loop has successfully run through every event in the DDOP, the user is notified as per Figure 4.16.

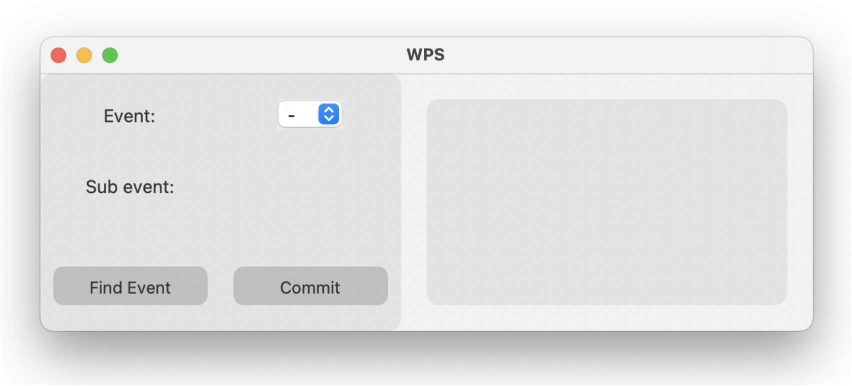


The engineering loop was successful. The plan is ready!

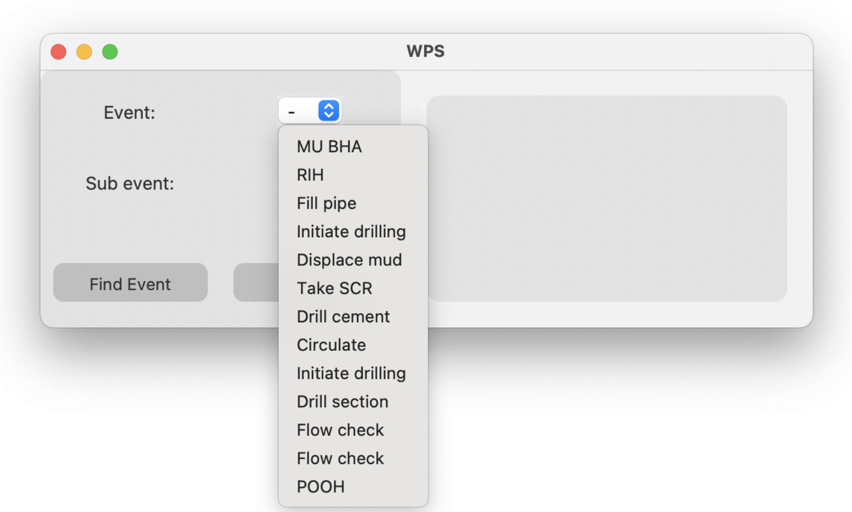
Figure 4.16: Engineering loop successful.

4.6 DDOP Automation module

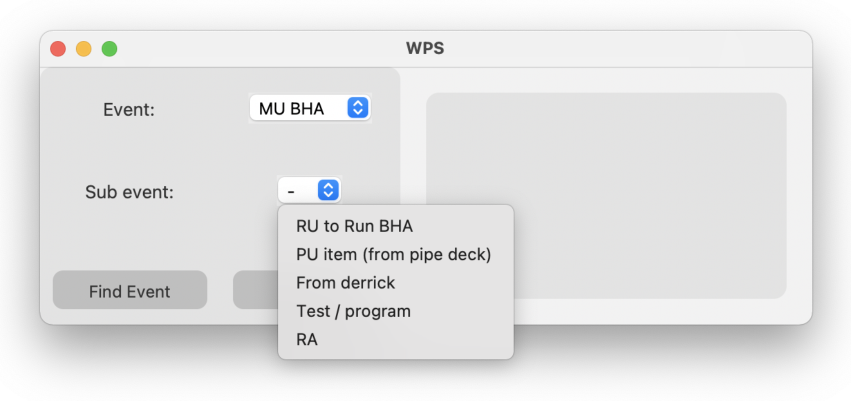
The automation module sequence developed to make the DDOP readable for RAS, is illustrated in Figure 4.17. The process is repeated for each line in the DDOP.



(a) Start window automation module.

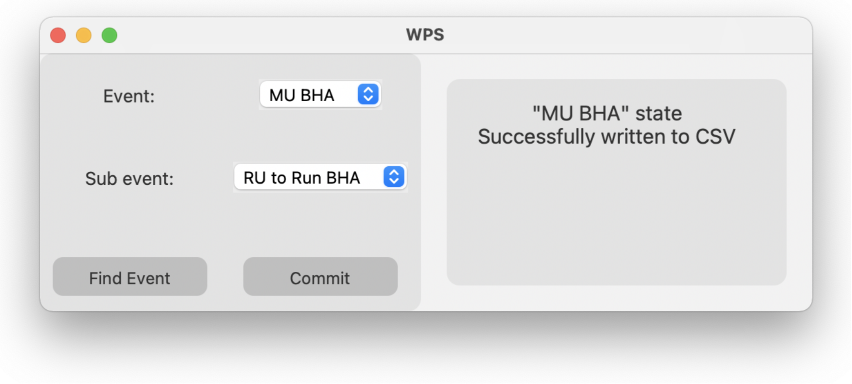


(b) User chooses event. Events are listed in the same order as in the DDOP.



(c) User chooses sub event. Choices in drop-down menu are based on event.

Figure 4.17: Automation module sequence.



Event: MU BHA

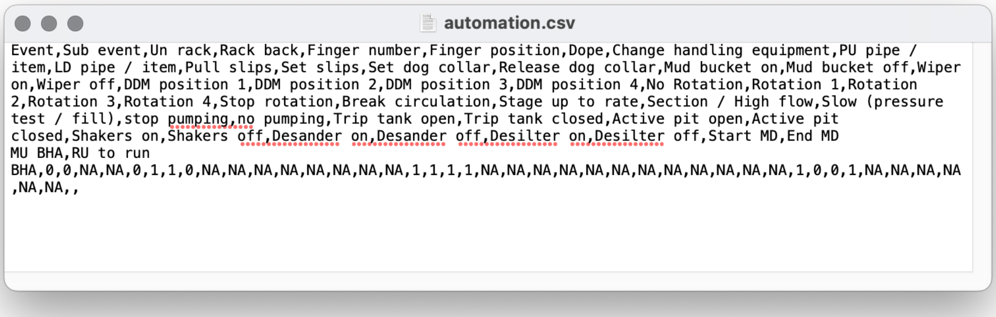
Sub event: RU to Run BHA

"MU BHA" state
Successfully written to CSV

(d) The user commits, and the event's states are written to CSV file.

Figure 4.17: Automation module sequence (cont.).

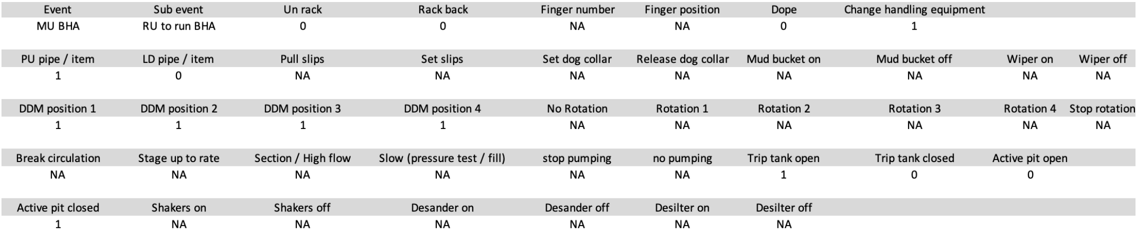
The results of the process shown in Figure 4.17 are stored in a CSV file, containing all the equipment that gets turned on/off. The first line in the CSV file can be seen in Figure 4.18. Here, each event, sub event and command is associated with a value. For event and sub event this is a text string, while for the commands this is "0", "1" or "NA". "NA" is used if the command does not apply to the sub-event in question.



```
Event,Sub event,Un rack,Rack back,Finger number,Finger position,Dope,Change handling equipment,PU pipe /
item,LD pipe / item,Pull slips,Set slips,Set dog collar,Release dog collar,Mud bucket on,Mud bucket off,Wiper
on,Wiper off,DDM position 1,DDM position 2,DDM position 3,DDM position 4,No Rotation,Rotation 1,Rotation
2,Rotation 3,Rotation 4,Stop rotation,Break circulation,Stage up to rate,Section / High flow,Slow (pressure
test / fill),stop pumping,no pumping,Trip tank open,Trip tank closed,Active pit open,Active pit
closed,Shakers on,Shakers off,Desander on,Desander off,Desilter on,Desilter off,Start MD,End MD
MU BHA,RU to run
BHA,0,0,NA,NA,0,1,1,0,NA,NA,NA,NA,NA,NA,NA,NA,1,1,1,1,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,1,0,0,1,NA,NA,NA,NA
,NA,NA,,
```

Figure 4.18: First line in the CSV file.

As Figure 4.18 is difficult to read for humans, Figure 4.19 illustrates the same file fixed into a normal spreadsheet. The CSV file for the full DDOP fixed into a spreadsheet can be found in Appendix D



Event	Sub event	Un rack	Rack back	Finger number	Finger position	Dope	Change handling equipment		
MU BHA	RU to run BHA	0	0	NA	NA	0	1		
PU pipe / item	LD pipe / item	Pull slips	Set slips	Set dog collar	Release dog collar	Mud bucket on	Mud bucket off	Wiper on	Wiper off
1	0	NA	NA	NA	NA	NA	NA	NA	NA
DDM position 1	DDM position 2	DDM position 3	DDM position 4	No Rotation	Rotation 1	Rotation 2	Rotation 3	Rotation 4	Stop rotation
1	1	1	1	NA	NA	NA	NA	NA	NA
Break circulation	Stage up to rate	Section / High flow	Slow (pressure test / fill)	stop pumping	no pumping	Trip tank open	Trip tank closed	Active pit open	
NA	NA	NA	NA	NA	NA	1	0	0	
Active pit closed	Shakers on	Shakers off	Desander on	Desander off	Desilter on	Desilter off			
1	NA	NA	NA	NA	NA	NA			

Figure 4.19: First line in the CSV file fixed in spreadsheet.

(Intentionally left blank.)

5 Discussion

5.1 WPS

The WPS presented in this thesis is a prototype. Thus, there are elements in the software that need further development. These elements are discussed in the following.

5.1.1 Work process

One of the objectives of this thesis was to identify an approach allowing the software to have its intended functionality. Due to the complexity of the task, a significant amount of time was spent to map system requirements and publicly available techniques and methods supporting the different functions and calculations. The majority of the work conducted in this thesis has been related to development of the software itself.

5.1.2 Engineering Module

The engineering module implemented in the WPS is only partly finished, yet it still provides insight to how engineering is directly integrated with the DDOP and the WPS. The loop is designed to illustrate different principles such as providing feedback and changing variables, rather than performing perfect calculations. Consequently, there is room for improvements.

Currently, the engineering module is implemented with simple hydraulic calculations. This does not cover all engineering disciplines in well planning, but, again, sufficient to illustrate the module's working principles. Moreover, the engineering module results are not saved when running trough the engineering loop. This means that calculations from other events are not taken into account during the engineering loop, as with the WOS. Implementing the variable map and the control system as per the WOS should be done in the next stage of development to mitigate this.

Another important note to make from the engineering module is that the process itself is not fully automated yet. In case of an error, the program will ask the user whether it would like to change equipment (BHA) or use a different mud weight. The WPS is designed this way to showcase how the different steps in the engineering loop. By implementing more logic and experience to the process, it can be automated and become ”smarter” and require less human interaction.

5.1.2.1 Prioritizing engineering

When establishing a well design, different types of engineering calculations may end up in conflict. For instance, should T&D be prioritized over pressure loss when designing casing program, or vice versa? A choice must be made based on what is more important in the specific case. Local factors applying to each well will influence the decision. The primary logic regarding the internal priority of the calculations, can follow the streamlined thinking of drilling engineers. On top, there are two functions designed to manipulate this:

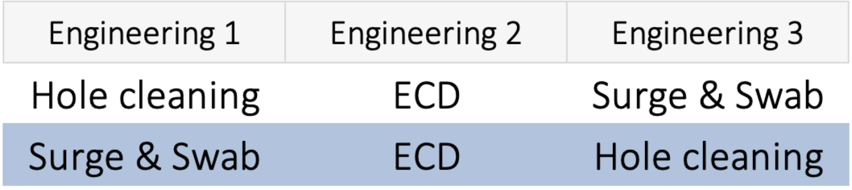
a) Digital experience

b) Manual setting by engineer

Human involvement is important in these kind of questions as they are responsible for the design - software can only support the process.

In automated well planning, decisions on which engineering is more important should be taken from the software, but with possibility for human override. This means that the software must know which one to prioritize. Of course, this can be hard coded resulting in a rigid solution, e.g. always prioritize T&D over pressure loss. Otherwise, the software must be designed in a way that it makes use of experiences and similar operations to take these decisions. This can also be done using machine learning where the software use previous data and well history to be able to make educated decisions.

The WPS has currently no functionality to prioritize one engineering calculation over another. However, it can be modified to solve this with a simple approach after the principle of "first come, first served". This means that the type of engineering that is listed first, is the one that is calculated first and has the highest priority (hard coded). Figure 5.1 illustrates priority between engineering. Engineering that is listed under "Engineering 1" have higher priority than engineering listed under "Engineering 2" or "Engineering 3". Each event has a maximum of three calculations associated, establishing the engineering priority for each event. By allowing the engineering that has the highest priority to be calculated first, should any other fail, the program can ask the user if they still want to continue seeing as the most important calculations were successful.



Engineering 1	Engineering 2	Engineering 3
Hole cleaning	ECD	Surge & Swab
Surge & Swab	ECD	Hole cleaning

Figure 5.1: Priority of engineering calculations.

5.1.3 Automation module

As previously mentioned, input to RAS is in this thesis represented with so-called states. Instructing equipment to be either *on* or *off* gives little room for individual adjustments. Additional information such as flow rate, mud weight and other drilling parameters must be provided to the RAS. As of today, this has not been included in the WPS and should be included in the next phase of development.

Furthermore, the process of "translating" the DDOP to a format compatible with RAS is currently a semi-manual process. The user must for every event choose correct sub event, making it prone to errors. As mentioned earlier, the level of detail is not the same for the DDOP and the DDOP automation module. Making the two modules even in terms of level of detail, i.e. adding one more layer of information to the DDOP, is straightforward yet time consuming and has been demonstrated in Section 3.4.1. Making the process fully automated when the level of detail match, is a simple task.

5.1.4 WPS as proof of concept for LCWIM and WOS

As stated in the start of this thesis, a main driver behind developing the WPS is to demonstrate concepts from the LCWIM and the WOS. Proving these concept is important for taking the development of the LCWIM to the next level.

The LCWIM and WOS is outlined in an object-oriented format, but not in an object-oriented programming format. The WPS is developed in an object-oriented format in Python. It illustrates several important aspects of object-orientation in programming. For instance, the WPS demonstrates how utilizing classes can be used to develop a working desktop application integrating multiple aspects of well planning. Within the class that is defined as the main application there are multiple functions. These functions control everything from buttons and entry-fields, opening new windows to controlling the engineering- and automation modules.

Moreover, the WPS integrates the well plan and engineering. The engineering loop runs through all events defined in the well plan without the use of other software. Integrating all elements in planning, including engineering, in the same platform is a key aspect in the LCWIM.

Enabling automated rig operations is also an important feature of the LCWIM. The solution presented in the WPS regarding making the DDOP readable for RAS, is a solution that can be adopted by the LCWIM.

The WPS utilizes a database to store and handle the DDOP. The database contains information about the well events in addition to contracts and equipment. Tying these elements together is a key aspect of the LCWIM and the WOS. The research and work done during this thesis lead the authors to believe that using databases is fitting also for the LCWIM.

On the flip side, it is important to keep in mind that the WPS is a small scale prototype. Principles and methods used in the WPS may not be applicable in a full scale application. As an example, making the WPS run locally is not in line with a full scale application integrating all parts in planning across disciplines. Today the standard is moving more and more towards cloud-based web applications, rather than desktop applications. Additionally, more specific elements such as using lists instead of classes is not ideal for a more comprehensive software where code structure and flexibility is key.

5.1.5 Challenges during the development of the WPS

Over the course of the development of the WPS, multiple challenges were encountered. Central challenges are summarized in the list below, and will be explained further:

1. Learning Python syntax and packages.
2. How to handle large quantities of data.
3. Choosing correct programming approach for the different parts of the application.

Both authors of the thesis have had limited experience with OOP in Python, and only some experience with OOP in similar programming languages such as C++. One of the main challenges in the first phase of the development was learning Python syntax and gain knowledge about the packages needed to develop the application. However, as presented in Appendix C, the language is straightforward and intuitive. Additionally, Python has an extensive set of learning materials available online making the process manageable. Using Python for the development of the WPS has not hindered the development in any way, shape or form. It is therefore believed that Python is fitting also for further development of the WPS and LCWIM.

Another challenge that arose during the development of the WPS was how data should be handled. There are multiple ways of handling data in Python, such as lists, dictionaries, databases, classes etc. The different methods have different characteristics. The way data should be stored is dictated by the type and format of the information that is being stored. The full DDOP is stored in a database due to the amount of information that is included, while the RAS-compatible DDOP is converted to a dictionary (before exported to a CSV file) to include equipment states in a simple manner.

To illustrate the difficulties of choosing the correct way to handle data, the challenge of picking the way to define the BHAs is a good exam. Several approaches were tried before choosing a method. Firstly, the BHA's were made such that each BHA-component was defined in its own class. A full BHA was then made up of several classes of BHA-components. However, this approach proved difficult to fully integrate into the engineering calculations. At the same time it is believed that this approach is possible and would account for an efficient and agile solution if successfully implemented. The approach that was used in the WPS in the end, was defining the BHA's in nested lists i.e. a list containing several other lists. This means that a BHA is made up of a list of components, which again contain lists of component properties. This allowed information such as OD and ID of each component to be extracted from the BHA and used in engineering calculations. The solution proved to be convenient for the purpose of testing the engineering loop, even though using classes would probably have been better.

Furthermore, choosing the correct programming approach for the different parts of the application was a challenge. The different parts of the program have different characteristics and requirements in terms of programming. For instance, the main menu was developed with focus on visual expression. Contrarily, the planner window was developed with focus on functionality rather than visuals. No previous experience from development of applications or programs equally sophisticated as the WPS, led to a significant amount of time were spent "trying and failing", before landing on one approach.

5.2 LCWIM

5.2.1 How the LCWIM design can influence today's industry

Oliasoft's WellDesign and Schlumberger's Delfi are two software systems providing well planning and support currently on the market. WellDesign, as introduced before, provide automatic well planning in a cloud-based environment. Delfi is an integrated software for the whole exploration and production value chain (Schlumberger, n.d.). The well planning process using Delfi is human driven, i.e. no automation of the planning and experiences depend on the involved personnel (Brechan, 2020).

As the industry is moving towards integrated software incorporating more aspects of well management, there is reason to believe that WellDesign, Delfi or similar software can be influenced by LCWIM. One of the biggest selling points with regards to the LCWIM is the focus on the entire life cycle of the well. With this focus, data containing information about a well from planning until P&A can be available in a single system integrating all disciplines. It is already evident that the industry is moving in this direction, as the upsides of automation and digitalization are too significant to ignore.

The LCWIM does not only focus on the engineering part of the life cycle of the well, but also on incorporation of administrative work directly into the system. Even though some of current well planning software, such as Landmark, Delfi and Oliasoft, include prices of projected wells, these does not take into consideration the specific contracts for actually drilling the well with all equipment, service companies etc. This is one of the main factors that separates today's well planning software from the LCWIM design.

Furthermore, engineering work will be more efficient by utilizing digital well planning technologies such as LCWIM. A common perception of automation and digitalization is that robots or data-systems would eventually replace people by performing tasks more efficiently and cheaply. To some extent, this is correct; but, by applying digitalization and automation, engineers in charge of well planning, completions, interventions, or P&A may devote more time to other vital areas of their jobs. For example, time spent on designing wells can be drastically reduced, making it possible to produce several designs for a single well and choose the most effective one. More time may also be spent on evaluating the quality of the designs and generating improved plans in terms of HSE.

5.2.2 Pitfalls using LCWIM and automation

As previously mentioned, an integrated system such as the LCWIM can provide a huge positive impact in terms of increasing efficiency, lowering workload and improving quality. However, one should also consider negative effects of using such a software. Firstly, extensive use of automation may harm general understanding of the planning process as tasks are performed by machines rather than humans. Referring to Appendix C.8, insight to the processes running the system will be important. Furthermore, less human involvement may harm quality of the final product. It is said that ”Computers are incredibly fast, accurate, but stupid. Humans are incredibly slow, inaccurate, but brilliant. Together they are powerful beyond imagination.” This proverb captures an important aspect of switching from human-oriented to a data-driven process such as the LCWIM: the importance of human involvement cannot be understated. The human ability to take decisions based on a life time of experience and ”gut feeling” is very hard to obtain using machines. Consequently, in case of moving to fully automated software systems, human involvement will still play a vital part.

It will be tempting for many profit driven companies to reduce their manpower as automation reduces the man hours for planning and operation of wells. However, the competence must be kept and maintained. Illustrative for just this is the autopilot on a plane. Under most circumstances, software can manage take-off and landing as well as maneuver the plane. Because software cannot be held accountable for errors that may occur, the pilot's role is to monitor and check the program's suggestions. Maintaining the competency of the pilot is therefore essential. Likewise, software and computers can do the footwork and assist the engineers in well planning, but not be responsible of the plans.

5.3 New technology enables new methods

5.3.1 Improved data democratization

As introduced in Section 2.1.1 every department has their own ”silo”, making planning a compartmentalized process. By moving to an integrated planning software data flow will be improved significantly. WellDesign, for instance, is updated continuously. This enables all involved parties to be up to date on the latest adjustment made to the well plan, saving both time and workload. Automation using intelligent software will let the wells team participate ”later” in planning. The subsurface team can perform many cheeks and make initial designs by themselves. The boarders between departments will be blurred.

5.3.2 Three-way safety system using RAS

Safety is priority number one when drilling. Using RAS provides an additional layer of safety in operations. The first layer of safety in operation is the engineers planning the wells. The plans are prepared, checked and quality assured. The driller follows this plan and takes action if unforeseen events happen or if needed, making it the second layer of safety. Furthermore, if RAS is implemented, this may inform the driller if something is wrong during the operation or take action itself, thus making it a third layer of security. The three-way safety system is illustrated in Figure 5.2.

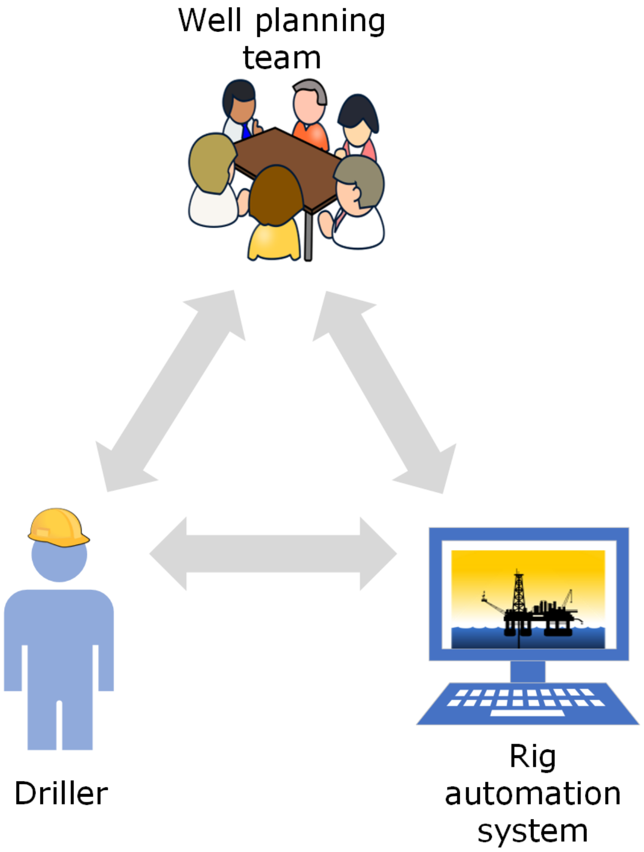


Figure 5.2: Three-way safety system.

The driller shall at all times have the highest priority in terms of ensuring the safety of the operation. This means that the driller can at all times shut down operations, even when RAS is used. RAS should provide operational support in terms of feedback and suggestions to actions, in addition to performing standardized drilling operations.

5.3.3 Cloud based software

Cloud-based software is becoming increasingly more popular. The technology is mature and available. Well planning applications have already been been moved to the cloud. Computation and storage take place in the cloud, rather than locally on every computer meaning that up-to-date data is available to everyone at all times. This is aligning with more integrated well planning operations across disciplines.

The LCWIM is intended to be cloud-based when fully developed. The WPS, however, is a program run locally in Python. Making the software available online was not one of the objectives of this thesis. However, Python is compatible with all cloud based environments the authors have investigated and moving the WPS, and ultimately the LCWIM, to the cloud is possible.

When using data driven technology, such as cloud technology, data security must have highest priority. Cyber-attacks are becoming increasingly more advanced and confidential data can end up in the wrong hands causing major repercussions. Cloud based software is considered safer than local ”silos”, as cloud services today provides encryption algorithms made so only authorized users can access data. As well as using encryption algorithms, cloud services also have cloud-based firewalls that provides a filter for malicious traffic as a virtual barrier around cloud platforms.

5.3.4 Machine learning (ML)

Machine learning is type of artificial intelligence that is a hot topic within the digital world today. It ”allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so” (Burns, 2021, §1). Machine learning algorithms learn from input data, making it able to predict outcomes and take decisions based on this. For instance, the suggestions that come up on Netflix are based on machine learning and what the user have watched previously. This kind of ”proposed solutions” are possible in well planning as well, using ML.

Having ”smart” software is a key to automated well planning. A standardized well plan is a good starting point for a well plan. However, adjustments must be made to the standardized plan to fit the objective and ensure the safety of each individual well. A software being able to accurately suggest and implement these changes will save a substantial amount of time and help optimizing well planning. With that in mind, machine learning will play an essential role in the future of well planning.

(Intentionally left blank.)

6 Conclusion

In this thesis a prototype for a well planning software, the WPS, has been developed and presented. Background theory for digitalization and automation in the petroleum industry as well as an introduction to important programming principles and elements is given. Additionally, the framework outlined by Brechan (2020) which the WPS is built on, is elaborated on. Further, the methodology of the WPS is explained, before the WPS was presented and discussed. The research conducted in this thesis has led to the following conclusions:

- A running prototype for well planning and creating DDOPs has been developed in Python, enabling an effective and agile way of establishing well plans.
- The prototype proves concepts of LCWIM and the WOS, such as integrating well plans directly to engineering, contracts and equipment.
- The prototype has been developed such that the DDOP is readable for humans as well as for RAS, enabling automated operations. The method used to achieve this is adoptable by the LCWIM.
- The programming approach used to develop the software is suitable for the intended functionality of the WPS. Using Python is believed to be fitting for further development of the WPS and LCWIM.
- The prototype demonstrates how object-oriented programming can be used to develop a well planning application, proving an important concept in the LCWIM.
- The majority of the work conducted in this thesis was spent on developing the software itself.
- The prototype contains a GUI for interaction with the user, a main menu for navigating the program and entering variables, a planning window with an interactive DDOP, an engineering module and an automation module.
- The engineering module implemented is fairly simple and not complete. At the same time, it illustrates how engineering is directly connected to the well plan.
- The WPS utilized databases to store and handle data, enabling easier data transfer between all disciplines.
- Some methods used in the WPS are not suitable to be scaled to a full size application, such as running the software locally.
- The prototype and LCWIM are compatible with cloud technology.
- Digitalization and automation lead to saved time, higher quality well plans and improved safety in operations. Additionally, departmentalization is reduced significantly.
- The WPS is a prototype and require further development in all aspects.

(Intentionally left blank.)

7 Further work

The research conducted in this thesis is part of a comprehensive model as described in Section 1.1. A lot of work still remain to prove the concept for all well planning. An application incorporating all aspects in well planning is possible, as is the idea behind the LCWIM. Important elements of further work can be summarized to:

- Develop WPS to also cover other parts of planning and well management than drilling.
- Develop the engineering module with all necessary calculations and implement variable map and control system as per the WOS.
- Improve the automation module. Order equipment state instructions to align with best practices and match the level of detail in the DDOP and the automation module.
- Add experiences from previous and similar operations. Develop program to incorporate this when establishing new well plans.
- Optimize code to improve computation time and improve efficiency.
- Incorporate ”smart” functionality such that the system can make decisions with minimal human interaction.
- Incorporate contracts and standardized equipment on a more detailed level than currently done.

References

Aadnoy, B. S. (2010). *Modern well design* (Second Edition). CRC Press.

Abrahamsen, E., Bergerud, R., Kluge, R. & King, M. (2015). Breakthrough in drilling automation saves rig time and safeguards against human error. *Abu Dhabi International Petroleum Exhibition and Conference*, Society of Petroleum Engineers.

Agarwal, H. (2022, January 28). *Polymorphism in c++*. GeeksForGeeks. Retrieved 21st April 2022, from https://www.geeksforgeeks.org/polymorphism-in-c/

Arkan, Ç. (2018, September 4). *From disruption to opportunity: How digital is transforming the future of oil and gas*. Microsoft. Retrieved 21st April 2022, from https://cloudblogs.microsoft.com/industry-blog/manufacturing/2018/09/04/from-disruption-to-opportunity-how-digital-is-transforming-the-future-of-oil-and-gas/

ASME Shale Shaker Committee. (2004). *Drilling fluids processing handbook*. Gulf Professional Publishing.

Bareksten, M. (2021). *Digital well planning* (tech. rep.). Norwegian University of Science and Technology.

BBC. (2019, October 5). *Monty python*. Retrieved 17th February 2022, from https://www.bbc.co.uk/mediacentre/latestnews/2019/monty-python-archive

Bourgoyne, A. T., Millheim, K. K., Chenevert, M. E., Young, F. S. et al. (1986). *Applied drilling engineering* (Vol. 2). Society of Petroleum Engineers.

Brechan, B. (2020). *Framework for automated well planning and digital well management* (Doctoral dissertation). Norwegian University of Science and Technology.

Brun, A., Trench, M. & Vermaat, T. (2017, October 31). *Why oil and gas companies must act on analytics*. McKinsey & Company. Retrieved 27th April 2022, from https://www.mckinsey.com/industries/oil-and-gas/our-insights/why-oil-and-gas-companies-must-act-on-analytics

Burns, E. (2021, March 30). *Machine learning*. TechTarget. Retrieved 5th May 2022, from https://www.techtarget.com/searchenterpriseai/definition/machine-learning-ML

De Wardt, J. P., Macpherson, J. D., Zamora, M., Dow, B., Hbaieb, S., Macmillan, R. A., Laing, M. L., DiFiore, A. M., Inabinett, C. E. & Anderson, M. W. (2015). Drilling systems automation roadmap-the means to accelerate adoption. *SPE/IADC Drilling Conference and Exhibition*, Society of Petroleum Engineers.

DiChristopher, T. (2015, March 5). *Oil firms are swimming in data they don't use*. CNBC. Retrieved 27th April 2022, from https://www.cnbc.com/2015/03/05/us-energy-industry-collects-a-lot-of-operational-data-but-doesnt-use-it.html

Doherty, E. (2020, April 15). *What is object-oriented programming? oop explained in depth*. Educative. Retrieved 12th March 2022, from https://www.educative.io/blog/object-oriented-programming#four

Future Learn. (2021, April 9). *What is python used for? 10 practical python uses*. Retrieved 17th February 2022, from https://www.futurelearn.com/info/blog/what-is-python-used-for

Gomez, E., Ombe, E., Goodkey, B. & Carvalho, R. (2021). Drilling automation: The step forward for improving safety, consistency, and performance in onshore gas drilling. *SPE Middle East Oil & Gas Show and Conference*, Society of Petroleum Engineers.

Hassan, M. A., Strand, J., Strømhaug, A. & Erichsen, L. L. (2022). 1-click automatic well design using integrated cloud software. *International Petroleum Technology Conference*, International Petroleum Technology Conference.

HMHW. (n.d.). *Automation*. Retrieved 7th April 2022, from https://hmhw.com/service/automation/

Kenton, W. (2022, March 6). *Black box model*. Investopedia. Retrieved 5th May 2022, from https://www.investopedia.com/terms/b/blackbox.asp

Lapeyrouse, N. J. (2002). Chapter five - engineering calculations. In N. J. Lapeyrouse (Ed.), *Formulas and calculations for drilling, production and workover* (Second Edition, pp. 165–207). Gulf Professional Publishing. https://doi.org/https://doi.org/10.1016/B978-075067452-2/50005-3

Matplotlib. (n.d.). *Matplotlib*. Retrieved 7th April 2022, from https://matplotlib.org

Mueller, J. (2018, August 27). *Embracing the four python programming styles*. New Relic. Retrieved 12th May 2022, from https://newrelic.com/blog/nerd-life/python-programming-styles

NOV. (n.d.). *Novos reflexive drilling system*. Retrieved 6th April 2022, from https://www.nov.com/products/novos

NOV. (2019). *Novos and wired drill pipe* (tech. rep.). NOV.

Oliasoft. (2022, April 8). *Oliasoft homepage*. Retrieved 1st May 2022, from https://www.oliasoft.com

Oracle. (n.d.). *What is a database?* Retrieved 5th April 2022, from https://www.oracle.com/database/what-is-database/

Padmanabhan, V. (2014, March 26). *Big data analysis in oil and gas*. Bain & Company. Retrieved 27th April 2022, from https://www.bain.com/insights/big-data-analytics-in-oil-and-gas/

pandas. (n.d.). *Pandas*. Retrieved 7th April 2022, from https://pandas.pydata.org

Peters, T. (2010). The zen of python. *Pro python* (pp. 301–302). Springer.

Pritchard, D., Roye, J. & Espinoza-Gala, L. (2012, November 1). *Real-time data offers critical tool to redefine well control, safety*. Drilling Contractor. Retrieved 26th April 2022, from https://www.drillingcontractor.org/real-time-data-offers-critical-tool-to-redefine-well-control-safety-19320

Python Software Foundation. (n.d.-a). *Python interface to tcl/tk*. Retrieved 6th April 2022, from https://docs.python.org/3/library/tkinter.html

Python Software Foundation. (n.d.-b). *Sqlite3 – db-api 2.0 interface for sqlite databases*. Retrieved 4th April 2022, from https://docs.python.org/3/library/sqlite3.html

Rystad Energy. (2019, October 8). *Oil industry can save $100 billion on digitalization*. Retrieved 19th April 2022, from https://www.rystadenergy.com/newsevents/news/press-releases/oil-industry-can-save-100-billion-on-digitalization/

Schimansky, T. (2022, June 1). *Customtkinter*. Github. Retrieved 21st April 2022, from https://github.com/TomSchimansky/CustomTkinter

Schlumberger. (n.d.). *Delfi cognutive ep environment*. Retrieved 12th May 2022, from https://www.software.slb.com/delfi

Scylla. (n.d.). *Differences between sql vs nosql*. Retrieved 5th April 2022, from https://www.scylladb.com/learn/nosql/nosql-vs-sql/

Tvedt, M. (2017). Experience transfer and risk analysis in digital well planning. *SPE Bergen One Day Seminar*, Society of Petroleum Engineers.

Vambheim, L. D. (2021). *Digital well planning* (tech. rep.). Norwegian University of Science and Technology.

Whitfield, S. (2017, July 9). *Human error: The big hurdle to safer facilities*. Journal of Petroleum Technology. Retrieved 12th May 2022, from https://jpt.spe.org/human-error-big-hurdle-safer-facilities

Ziatdinov, S. & Thomas Philip, T. (2021). Step change transformation of legacy rigs to autonomous drilling rigs. *Abu Dhabi International Petroleum Exhibition & Conference*, Society of Petroleum Engineers.

(Intentionally left blank.)

Appendix

A Databases

A database is an organized collection of structured information, or data, typically stored in a computer system (Oracle, n.d.). There are different database systems today, with the most commonly used being relational databases. As data is stored in column and rows, there is a resemblance to excel spreadsheets. However, there are differences between the two in terms of how to access, store and manipulate data (Oracle, n.d.). Today, most relational databases use SQL to query, manipulate and defining data within databases (Oracle, n.d.).

Relational databases are structural databases with a defined set of columns and rows for each defined table. On the other hand, there are also NoSQL (Not only Structured Query Language) databases called non-relational databases. Non-relational databases became more popular during the 1990's after the rise of the internet and the need for a fast and reliable way to store non-structured data (Oracle, n.d.).

Figure A.1 shows the main differences between and SQL and a NoSQL database. A SQL database, also called a relational database management system is governed by table-like structures, where some properties can link one table to another. A NoSQL database, does not have this relationship between all of the entries but rather consists of key-value pairs and documents linked to each pair.

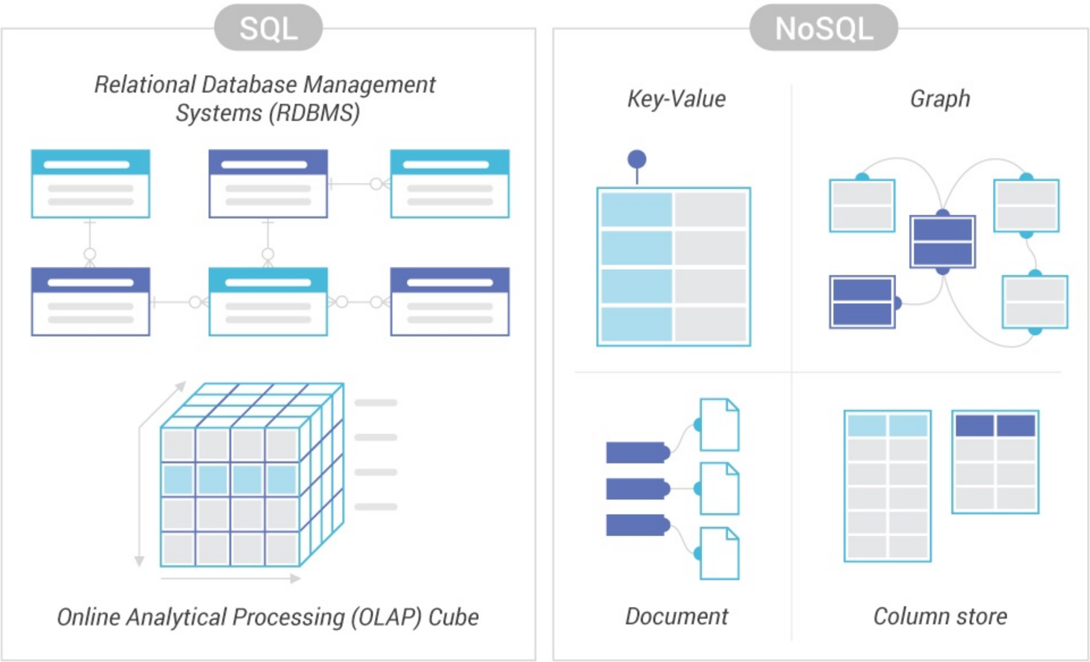


Figure A.1: Difference between SQL and NoSQL databases illustrated (Scylla, n.d.).

Creating the WPS, a SQL database serves the best purpose for storing data. That is because each entry that has to be stored from the DDOP follows a table-structure with a given number of rows and columns that needs to be filled in. There are plenty SQL databases to decide upon when first creating a database. Most commonly, the databases are stored on a server and not locally on your machine. For prototyping and smaller databases, however, the built-in Python library SQLite3 is a good choice. SQLite3 will be further discussed in Section C.1.

B Object-oriented programming

The WPS presented in this thesis uses an object-oriented programming (OOP) approach. There exists four different types of programming approaches today. These are summarized below (Mueller, 2018):

- Functional: Program built using functions treated as mathematical expressions.
- Imperative: Program built using computations to directly change the program's state.
- Procedural: Program built where tasks are dealt with step-by-step, calling functions as they are needed.
- Object-oriented: Program built using classes and objects with predefined functions only relative to themselves.

OOP utilizes classes and objects to create programs. Specifically, classes are used to define instances of objects. Objects are abstract examples of the classes, created with specific data (Doherty, 2020). As an example one can imagine having a class Human. This class has attributes that all humans have such as name, age and gender. From this class, one can create an object called *Person_1*, that is given the mentioned attributes values such as 'Peter', 24, and 'male'. Each object that is created from a class can be given different attribute values. The objects created can be referenced and used later in the program.

OOP is a widely know programming paradigm and include languages such as Python, Java and C++. Using OOP has several advantages (Doherty, 2020):

- Complex things are modeled as reproducible, simple structures.
- It produces reusable code that can be used across programs.
- Easy to debug as classes often contain all applicable information to them.
- Facilitates class specific behavior through polymorphism.
- Provides secure programs through encapsulation.

The advantages of OOP is a direct result of application of four fundamental principles tabulated in Table B.1.

Table B.1: The four principles of object-oriented programming.

Inheritance	Allows functions and attributes/variables to be inherited from one class to another. This allows for reusing code, which enhance readability and minimizes risk of mistakes.
Encapsulation	Confining code within objects, and only expose selected information outside of the class (Doherty, 2020). It is done to increase the overall security of the code, as only certain information can be extracted and used outside the class itself.
Abstraction	Using simple classes to represent complexity (Doherty, 2020). This is done to simplify the interaction between the user and the program.
Polymorphism	Polymorphism is the ability for something to be displayed in multiple forms (Agarwal, 2022). For instance, a function can be defined many times with different types of input (integer, double, etc.). The type used when calling the function will then govern which ”version” of the function that is used.

C Python

Python is a widely used high level programming language. The language is used for different purposes such as web and software design, data analytics and data visualization (Future Learn, 2021). It emphasises readability and aims to be a fun language to use. In fact, the language is named after the British comedy group Monty Python, see Figure C.1. The syntax in Python is often referred to as simple and easy to read. In the collection "Zen of Python", software developer Tim Peters outlined 19 guiding principles that reflect Python's main philosophy (Peters, 2010). Five of these are listed below:

- Beautiful is better than ugly.
- Explicit is better than implicit.
- Simple is better than complex.
- Complex is better than complicated.
- Readability counts.

Figure C.1: Python is named after the British comedy group Monty Python (BBC, 2019).

Python has an extensive collection of libraries and packages made to serve different purposes. Simplified, these are pre-made code allowing for development sophisticated programs using simple commands. In the following, an introduction to libraries and packages that have been used to develop the WPS will be presented.

C.1 SQLite3

As described in Appendix A, the WPS is using a SQL relational database for storing data, with the built in SQLite3 library. The python documentation of SQLite3 describes the library as the following (Python Software Foundation, n.d.-b, §1):

> SQLite3 is a C library that provides a lightweight disk-based database that doesn't require a separate server process and allows accessing the database using a nonstandard variant of the SQL query language. Some applications can use SQLite3 for internal data storage. It's also possible to prototype an application using SQLite and then port the code to a larger database such as PostegreSQL or Oracle.

Similar to other programming languages, SQL follows a clear syntax for querying, modifying and managing databases and tables within each database. Listing C.1 provides a short example of how to create and alter a table using SQLite3.

```python
# Import packages
import sqlite3

def create_and_insert():
    # Create a connection to the database
    conn = sqlite3.connect('test_database.db')
    # Create a cursor object
    c = conn.cursor()

    # Create a new table if it doesn't exist already
    c.execute("""CREATE TABLE if not exists test_table (
        height int,
        age int)
        """)

    # Enter data into the table
    c.execute("INSERT INTO test_table VAULUES (180,25)")

    # Commit the changes
    conn.commit()

    # Close the connection to the database
    conn.close()
```

Listing C.1: Example of SQL in SQLite3.

C.2 NumPy

NumPy is short for ”Numerical Python” and is a package used for scientific computations within python. Specifically, NumPy is used to create and work with arrays, as well as introducing other mathematical concepts such as trigonometric functions.

C.3 Pandas

Pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language (pandas, n.d.). The library can be used to load structured data from other sources, such as SQL tables, excel spreadsheets or comma-separated values (CSV) files, into a two dimensional array called DataFrame. When data is loaded into the DataFrame, Pandas is used for manipulating or analyzing the data.

C.4 Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible (Matplotlib, n.d.). One of the modules of this library is pyplot. The pyplot module is an interface of interacting with Matplotlib, in a MATLAB-like syntax. This makes plotting with Matplotlib simpler, introducing known plotting functions such as plot, scatter and bar.

Listing C.2 is made to illustrate the simplicity of Matplotlib and pyplot, as well as showing some basic functions of NumPy and Pandas. First Pandas is used to import a excel sheet with x-values ranging from -2π to 2π, and a constant y-value. Then the y-values are changed using NumPy, setting the y-values to $y = \sin x$. The values are then plotted against each other using pyplot, which can be seen in Figure C.2.

```
# Importing Packages
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Using pandas to extract the columns named X and Y from example.xlsx
example = pd.read_excel(r'excel/example.xlsx')
example_data = pd.DataFrame(example)
example_x = example_data.iloc[:]['X']
example_y = example_data.iloc[:]['Y']

# Using numpy to alter data
example_y = np.sin(example_x)

#Using matplotlib.pyplot to plot X and Y
figure = plt.figure("Sin")
title = plt.title("y = sin(x)")
plt.plot(example_x,example_y,'blue')
plt.show()
```

Listing C.2: Example of NumPy, Pandas and Matplotlib.

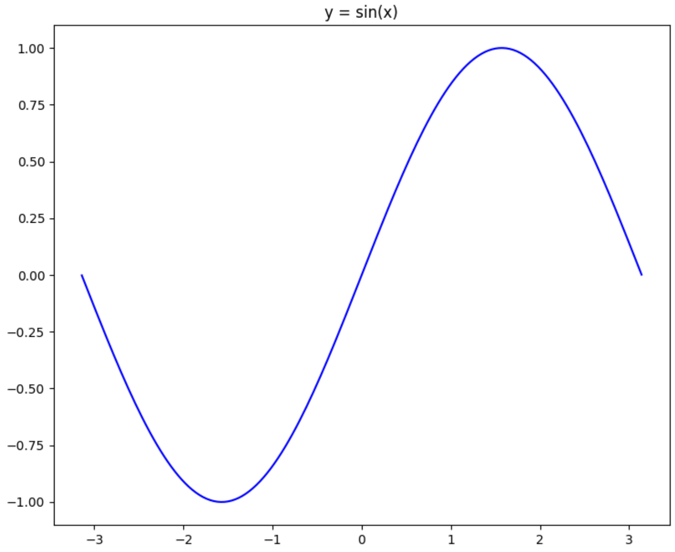


Figure C.2: Sinus plot created using NumPy, Pandas and Matplotlib.

C.5 Pillow

Pillow is a Python Imaging Library (PIL), used to process images in Python.

C.6 CSV

CSV is a file format widely used to store data. The package is needed for writing, reading and editing CSV files within a Python script.

C.7 Tkinter

Tkinter is a package used to interact with the Tk/Tcl toolkit, which is built in to Python. The Tk/Tcl toolkit provides a fast, reliable and stable way of displaying a graphical user interface (GUI). The hierarchy of Tk/Tcl is explained as the following (Python Software Foundation, n.d.-a, §6-8):

- Tcl: a dynamic interpreted programming language, just like Python. Though it can be used on its own as a general-purpose programming language, it is most commonly embedded into C applications as a scripting engine or an interface to the Tk toolkit.
- Tk: a Tcl package implemented in C that adds custom commands to create and manipulate GUI widgets. Each Tk object embeds its own Tcl interpreter instance with Tk loaded into it.
- Themed Tk (Ttk): a modern version of Tk widgets that provide a much better appearance on different platforms than many of the classic Tk widgets.

Tk and Tcl has been an integral part of Python for a long time (Python Software Foundation, n.d.-a), built to be a fast and reliable way for displaying and interacting with windows containing widgets like buttons, radio buttons, text-boxes etc.

Within the Tkinter package there are several ways of modifying how widgets look. An example is Tkinter themed which is built-in to Tkinter giving more modern-looking widgets then the original package. Another method of customizing how widgets look are by using customized Tkinter packages, such as CTkinter by Schimansky (2022), used in the WPS. These are often available as open source code online.

Listing C.3 produces a window with widgets created using three different packages. The window can be seen in Figure C.3 in both dark and light mode on MacOS.

```
from tkinter import *
import tkinter as tk
from tkinter import ttk
import customtkinter as ctk

class Window():
    def __init__(self,root):
        self.root = root
        self.root.geometry("160x200")

        self.frame = tk.Frame(self.root)
        self.frame.grid(row=0,column=0, sticky='nswe')

        self.button_tk = tk.Button(text="tk Button")
        self.button_tk.grid(row=0,column=0,pady=20,padx=20)

        self.button_ttk = ttk.Button(text="ttk Button")
        self.button_ttk.grid(row=1,column=0,pady=20,padx=20)

        self.button_ctk = ctk.CTkButton(text="ctk Button")
        self.button_ctk.grid(row=2,column=0,pady=20,padx=20)

app = Window(tk.Tk())
app.root.mainloop()
```

Listing C.3: Code producing Tkinter (Tk)-, Themed Tkinter (TTk)- and Custom-Tkinter (CTk) buttons.

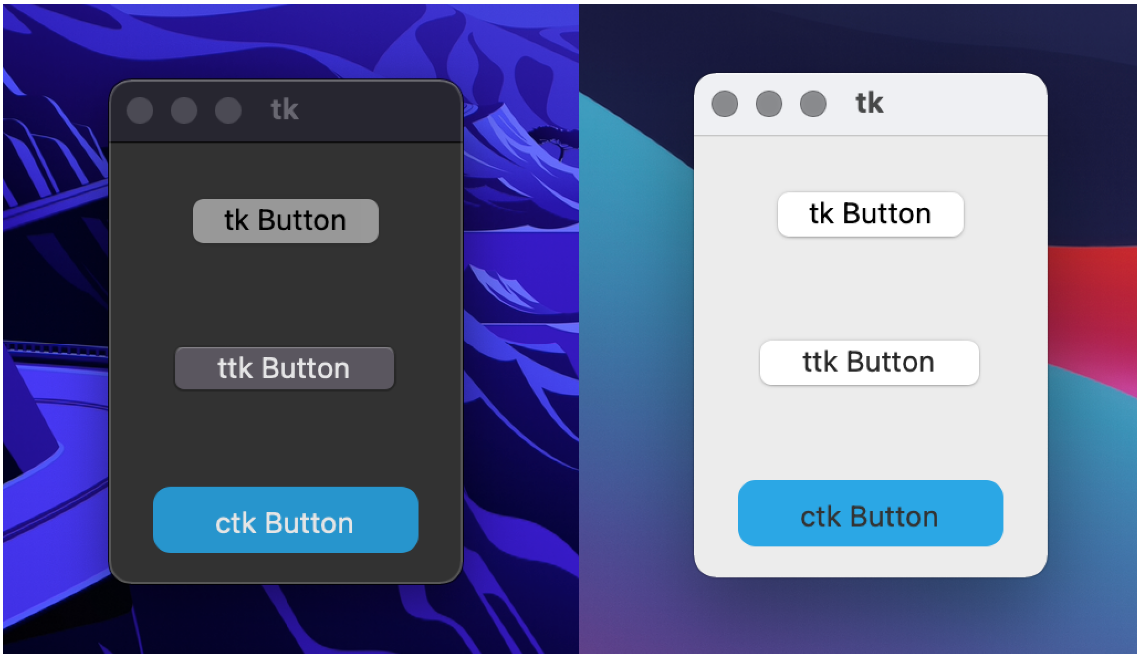


Figure C.3: Difference between buttons in Tkinter (Tk), Themed Tkinter (TTk) and CustomTkinter (CTk).

C.8 Black Box

”In science, computing, and engineering, a black box is a device, system, or object which produces useful information without revealing any information about its internal workings. The explanations for its conclusions remain opaque or ”black”” (Kenton, 2022, §1). Figure C.4 illustrates a black box. Here, the user feeds input to the system and receives output directly. What happens inside the system is unknown.

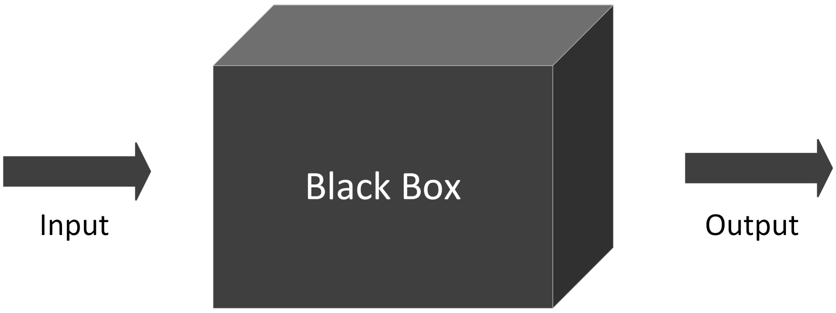
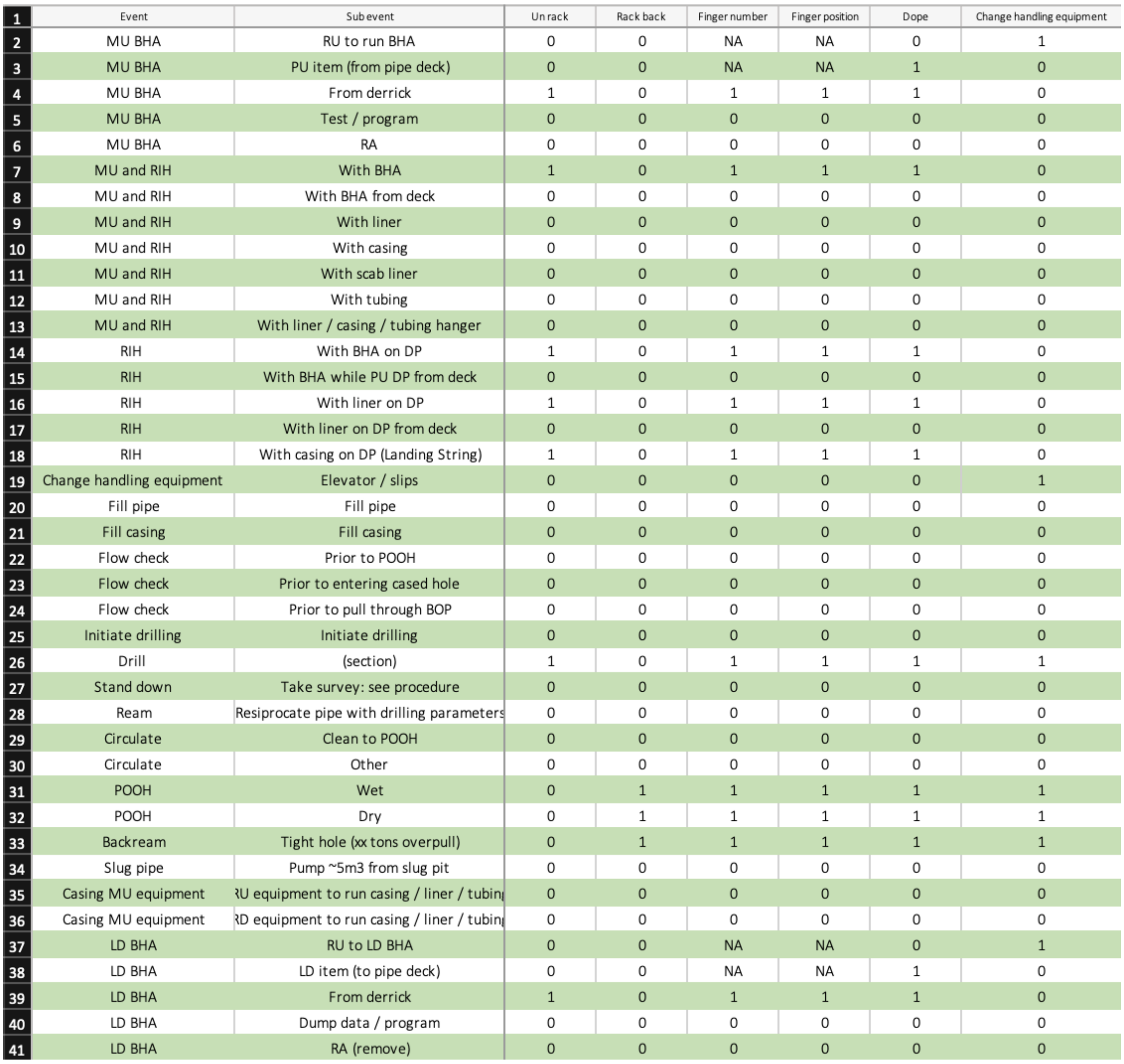


Figure C.4: Black Box Model.

Black boxes are generally considered to be negative when talking about digitalization and automation. When using software such as the WPS and other planning tools, it is important to know about the engineering principles working behind the application interface. Only working with input and output in a model without extensive knowledge about the working processes, will harm the quality of final product. Software functionality where formulas and calculations can be viewed in full upon request, can mitigate this.

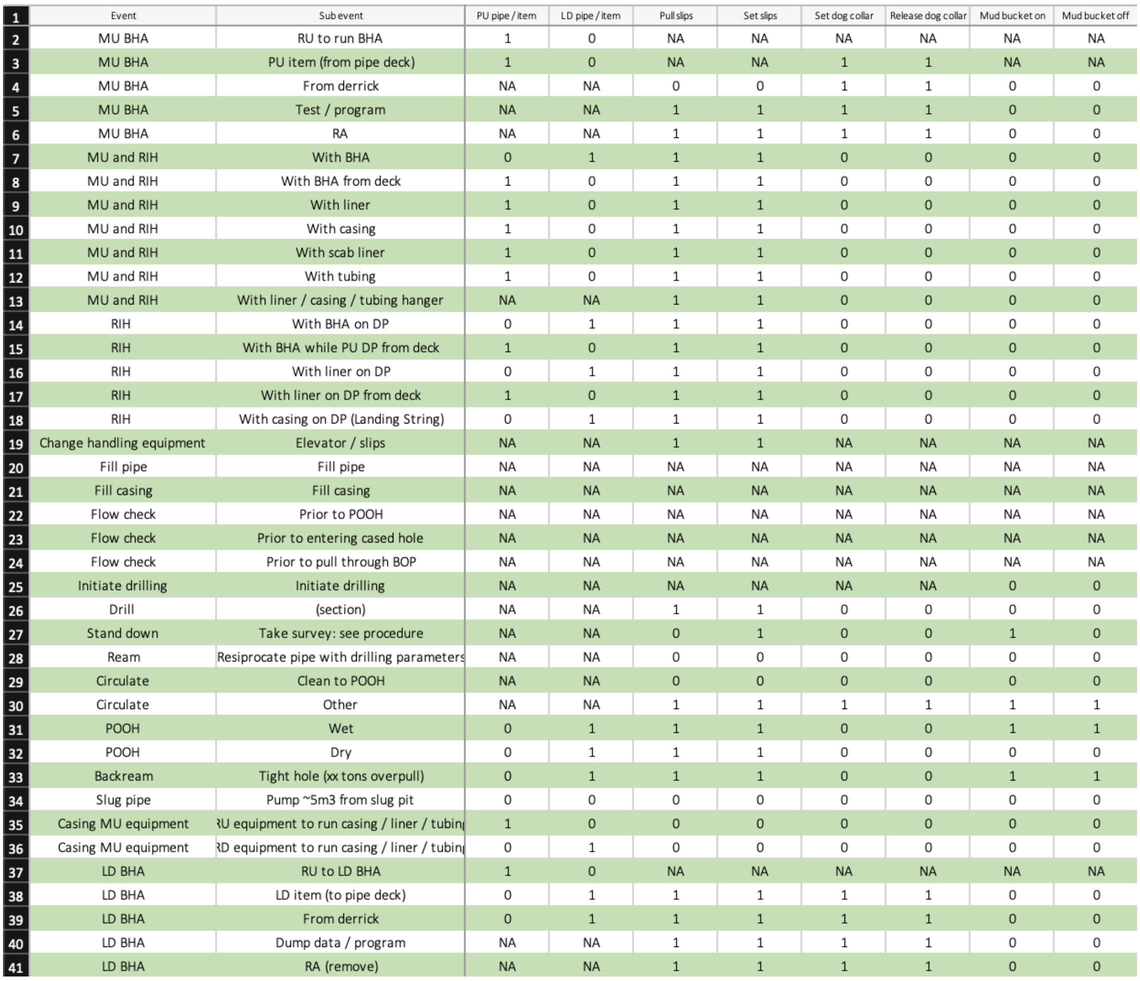
D Full CSV file from automation module in the planner window

The full CSV file from the planner window is shown in Figure D.1, D.2, D.3, D.4, D.5 and D.6.



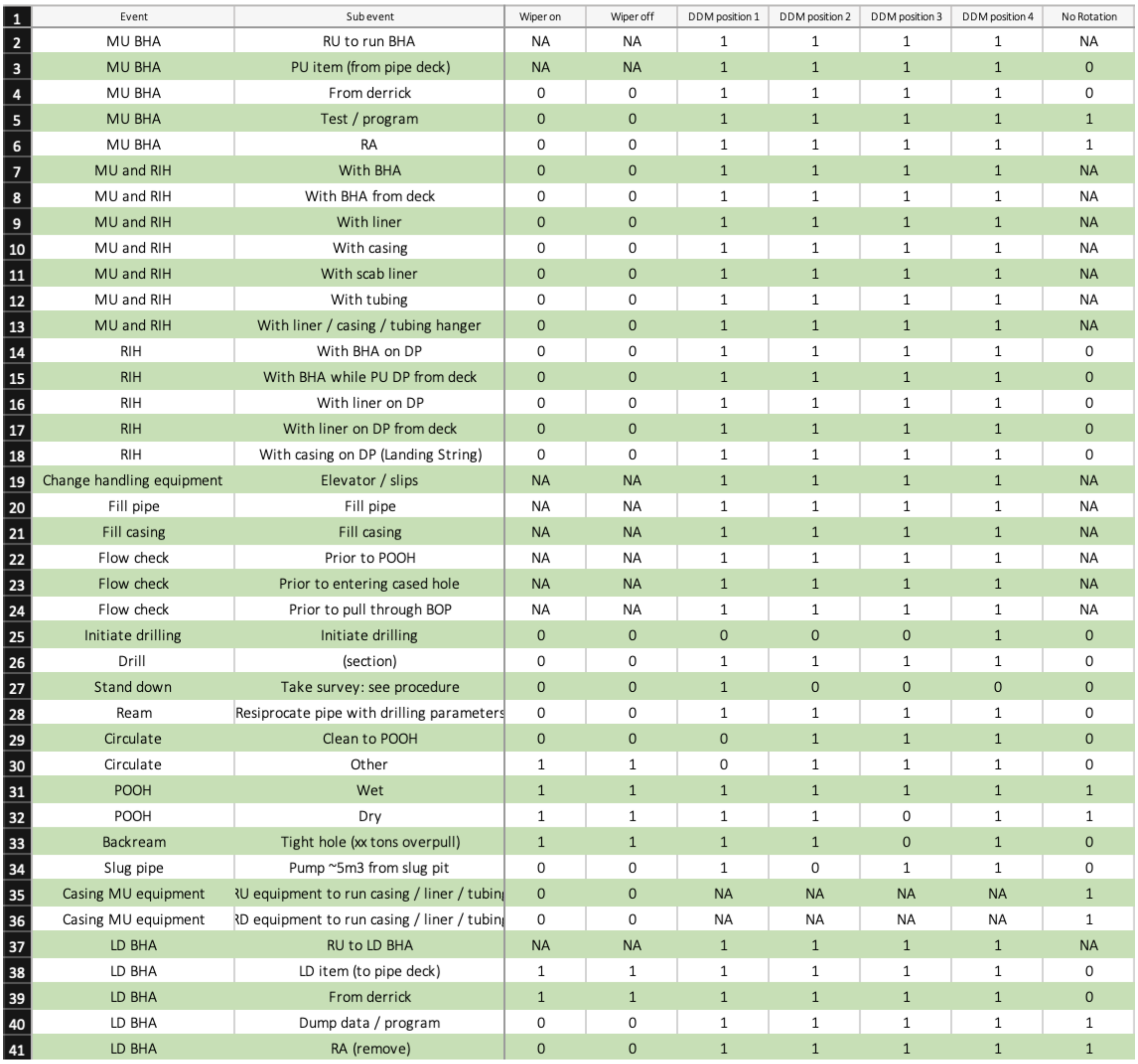
1	Event	Sub event	Un rack	Rack back	Finger number	Finger position	Dope	Change handling equipment
2	MU BHA	RU to run BHA	0	0	NA	NA	0	1
3	MU BHA	PU item (from pipe deck)	0	0	NA	NA	1	0
4	MU BHA	From derrick	1	0	1	1	1	0
5	MU BHA	Test / program	0	0	0	0	0	0
6	MU BHA	RA	0	0	0	0	0	0
7	MU and RIH	With BHA	1	0	1	1	1	0
8	MU and RIH	With BHA from deck	0	0	0	0	0	0
9	MU and RIH	With liner	0	0	0	0	0	0
10	MU and RIH	With casing	0	0	0	0	0	0
11	MU and RIH	With scab liner	0	0	0	0	0	0
12	MU and RIH	With tubing	0	0	0	0	0	0
13	MU and RIH	With liner / casing / tubing hanger	0	0	0	0	0	0
14	RIH	With BHA on DP	1	0	1	1	1	0
15	RIH	With BHA while PU DP from deck	0	0	0	0	0	0
16	RIH	With liner on DP	1	0	1	1	1	0
17	RIH	With liner on DP from deck	0	0	0	0	0	0
18	RIH	With casing on DP (Landing String)	1	0	1	1	1	0
19	Change handling equipment	Elevator / slips	0	0	0	0	0	1
20	Fill pipe	Fill pipe	0	0	0	0	0	0
21	Fill casing	Fill casing	0	0	0	0	0	0
22	Flow check	Prior to POOH	0	0	0	0	0	0
23	Flow check	Prior to entering cased hole	0	0	0	0	0	0
24	Flow check	Prior to pull through BOP	0	0	0	0	0	0
25	Initiate drilling	Initiate drilling	0	0	0	0	0	0
26	Drill	(section)	1	0	1	1	1	1
27	Stand down	Take survey: see procedure	0	0	0	0	0	0
28	Ream	Resiprocate pipe with drilling parameters	0	0	0	0	0	0
29	Circulate	Clean to POOH	0	0	0	0	0	0
30	Circulate	Other	0	0	0	0	0	0
31	POOH	Wet	0	1	1	1	1	1
32	POOH	Dry	0	1	1	1	1	1
33	Backream	Tight hole (xx tons overpull)	0	1	1	1	1	1
34	Slug pipe	Pump ~5m3 from slug pit	0	0	0	0	0	0
35	Casing MU equipment	RU equipment to run casing / liner / tubin	0	0	0	0	0	0
36	Casing MU equipment	RD equipment to run casing / liner / tubin	0	0	0	0	0	0
37	LD BHA	RU to LD BHA	0	0	NA	NA	0	1
38	LD BHA	LD item (to pipe deck)	0	0	NA	NA	1	0
39	LD BHA	From derrick	1	0	1	1	1	0
40	LD BHA	Dump data / program	0	0	0	0	0	0
41	LD BHA	RA (remove)	0	0	0	0	0	0

Figure D.1: Full CSV-file (1/6).



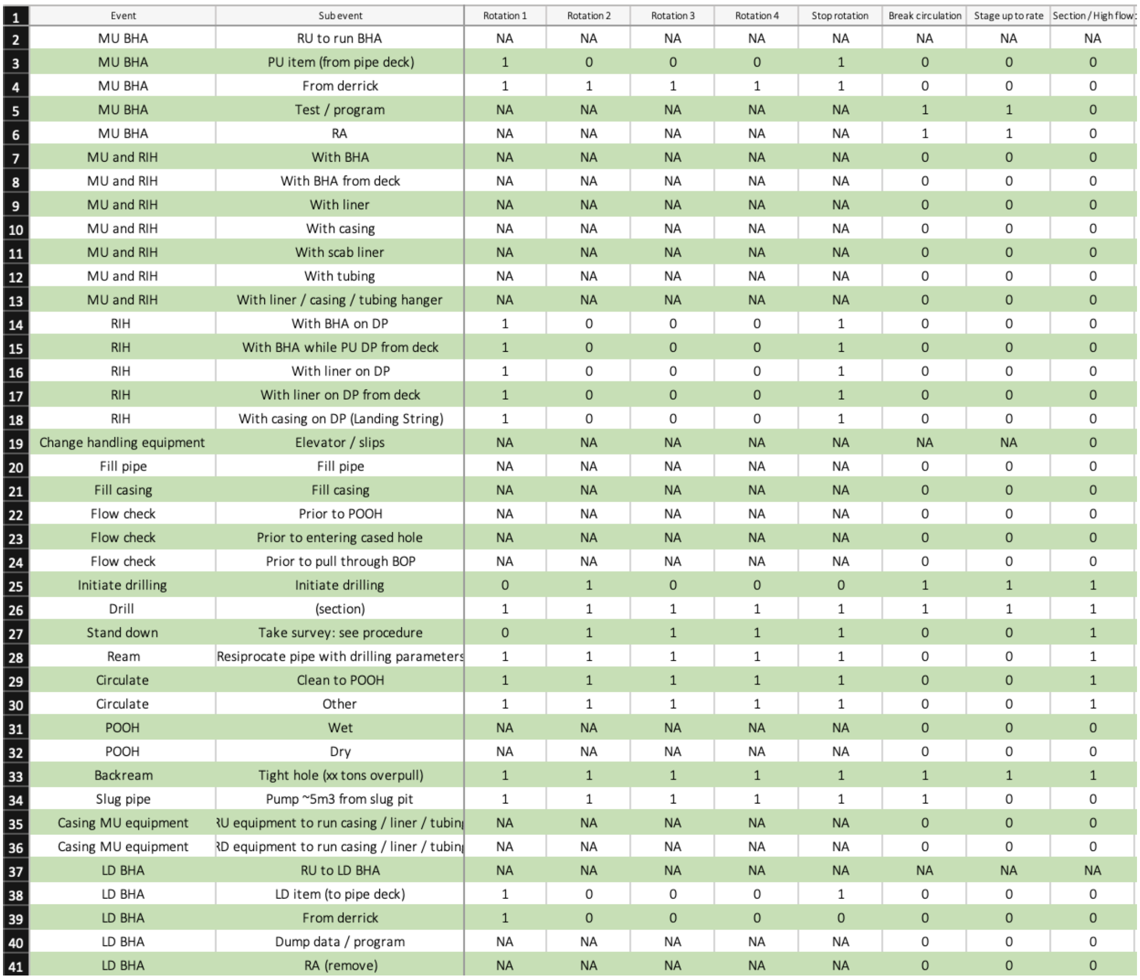
1	Event	Sub event	PU pipe / item	LD pipe / item	Pull slips	Set slips	Set dog collar	Release dog collar	Mud bucket on	Mud bucket off
2	MU BHA	RU to run BHA	1	0	NA	NA	NA	NA	NA	NA
3	MU BHA	PU item (from pipe deck)	1	0	NA	NA	1	1	NA	NA
4	MU BHA	From derrick	NA	NA	0	0	1	1	0	0
5	MU BHA	Test / program	NA	NA	1	1	1	1	0	0
6	MU BHA	RA	NA	NA	1	1	1	1	0	0
7	MU and RIH	With BHA	0	1	1	1	0	0	0	0
8	MU and RIH	With BHA from deck	1	0	1	1	0	0	0	0
9	MU and RIH	With liner	1	0	1	1	0	0	0	0
10	MU and RIH	With casing	1	0	1	1	0	0	0	0
11	MU and RIH	With scab liner	1	0	1	1	0	0	0	0
12	MU and RIH	With tubing	1	0	1	1	0	0	0	0
13	MU and RIH	With liner / casing / tubing hanger	NA	NA	1	1	0	0	0	0
14	RIH	With BHA on DP	0	1	1	1	0	0	0	0
15	RIH	With BHA while PU DP from deck	1	0	1	1	0	0	0	0
16	RIH	With liner on DP	0	1	1	1	0	0	0	0
17	RIH	With liner on DP from deck	1	0	1	1	0	0	0	0
18	RIH	With casing on DP (Landing String)	0	1	1	1	0	0	0	0
19	Change handling equipment	Elevator / slips	NA	NA	1	1	NA	NA	NA	NA
20	Fill pipe	Fill pipe	NA	NA	NA	NA	NA	NA	NA	NA
21	Fill casing	Fill casing	NA	NA	NA	NA	NA	NA	NA	NA
22	Flow check	Prior to POOH	NA	NA	NA	NA	NA	NA	NA	NA
23	Flow check	Prior to entering cased hole	NA	NA	NA	NA	NA	NA	NA	NA
24	Flow check	Prior to pull through BOP	NA	NA	NA	NA	NA	NA	NA	NA
25	Initiate drilling	Initiate drilling	NA	NA	NA	NA	NA	NA	0	0
26	Drill	(section)	NA	NA	1	1	0	0	0	0
27	Stand down	Take survey: see procedure	NA	NA	0	1	0	0	1	0
28	Ream	Resiprocate pipe with drilling parameters	NA	NA	0	0	0	0	0	0
29	Circulate	Clean to POOH	NA	NA	0	0	0	0	0	0
30	Circulate	Other	NA	NA	1	1	1	1	1	1
31	POOH	Wet	0	1	1	1	0	0	1	1
32	POOH	Dry	0	1	1	1	0	0	0	0
33	Backream	Tight hole (xx tons overpull)	0	1	1	1	0	0	1	1
34	Slug pipe	Pump ~5m3 from slug pit	0	0	0	0	0	0	0	0
35	Casing MU equipment	RU equipment to run casing / liner / tubin	1	0	0	0	0	0	0	0
36	Casing MU equipment	RD equipment to run casing / liner / tubin	0	1	0	0	0	0	0	0
37	LD BHA	RU to LD BHA	1	0	NA	NA	NA	NA	NA	NA
38	LD BHA	LD item (to pipe deck)	0	1	1	1	1	1	0	0
39	LD BHA	From derrick	0	1	1	1	1	1	0	0
40	LD BHA	Dump data / program	NA	NA	1	1	1	1	0	0
41	LD BHA	RA (remove)	NA	NA	1	1	1	1	0	0

Figure D.2: Full CSV-file (2/6).



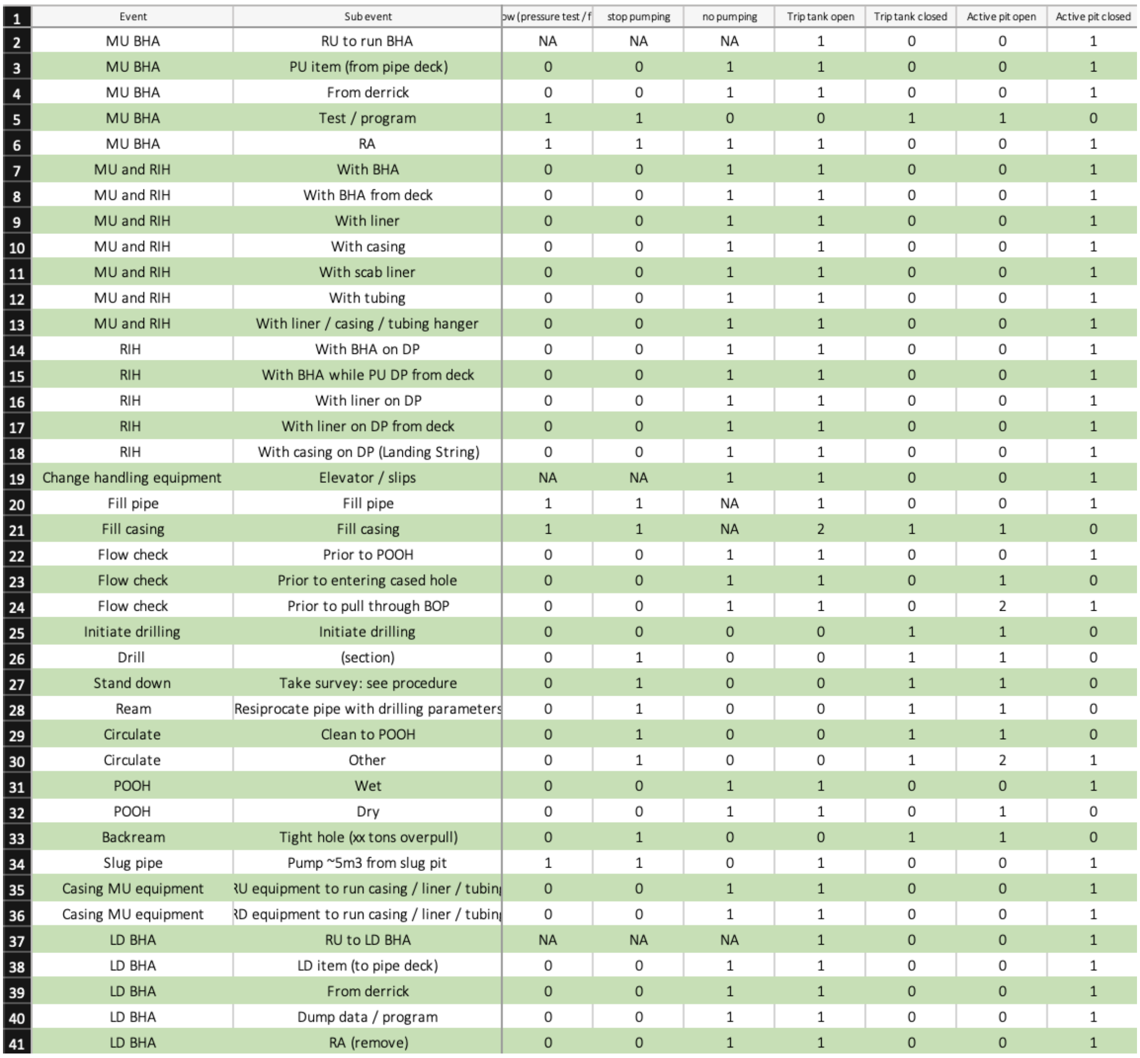
1	Event	Sub event	Wiper on	Wiper off	DDM position 1	DDM position 2	DDM position 3	DDM position 4	No Rotation
2	MU BHA	RU to run BHA	NA	NA	1	1	1	1	NA
3	MU BHA	PU item (from pipe deck)	NA	NA	1	1	1	1	0
4	MU BHA	From derrick	0	0	1	1	1	1	0
5	MU BHA	Test / program	0	0	1	1	1	1	1
6	MU BHA	RA	0	0	1	1	1	1	1
7	MU and RIH	With BHA	0	0	1	1	1	1	NA
8	MU and RIH	With BHA from deck	0	0	1	1	1	1	NA
9	MU and RIH	With liner	0	0	1	1	1	1	NA
10	MU and RIH	With casing	0	0	1	1	1	1	NA
11	MU and RIH	With scab liner	0	0	1	1	1	1	NA
12	MU and RIH	With tubing	0	0	1	1	1	1	NA
13	MU and RIH	With liner / casing / tubing hanger	0	0	1	1	1	1	NA
14	RIH	With BHA on DP	0	0	1	1	1	1	0
15	RIH	With BHA while PU DP from deck	0	0	1	1	1	1	0
16	RIH	With liner on DP	0	0	1	1	1	1	0
17	RIH	With liner on DP from deck	0	0	1	1	1	1	0
18	RIH	With casing on DP (Landing String)	0	0	1	1	1	1	0
19	Change handling equipment	Elevator / slips	NA	NA	1	1	1	1	NA
20	Fill pipe	Fill pipe	NA	NA	1	1	1	1	NA
21	Fill casing	Fill casing	NA	NA	1	1	1	1	NA
22	Flow check	Prior to POOH	NA	NA	1	1	1	1	NA
23	Flow check	Prior to entering cased hole	NA	NA	1	1	1	1	NA
24	Flow check	Prior to pull through BOP	NA	NA	1	1	1	1	NA
25	Initiate drilling	Initiate drilling	0	0	0	0	0	1	0
26	Drill	(section)	0	0	1	1	1	1	0
27	Stand down	Take survey: see procedure	0	0	1	0	0	0	0
28	Ream	Resiprocate pipe with drilling parameters	0	0	1	1	1	1	0
29	Circulate	Clean to POOH	0	0	0	1	1	1	0
30	Circulate	Other	1	1	0	1	1	1	0
31	POOH	Wet	1	1	1	1	1	1	1
32	POOH	Dry	1	1	1	1	0	1	1
33	Backream	Tight hole (xx tons overpull)	1	1	1	1	0	1	0
34	Slug pipe	Pump ~5m3 from slug pit	0	0	1	0	1	1	0
35	Casing MU equipment	RU equipment to run casing / liner / tubin	0	0	NA	NA	NA	NA	1
36	Casing MU equipment	RD equipment to run casing / liner / tubin	0	0	NA	NA	NA	NA	1
37	LD BHA	RU to LD BHA	NA	NA	1	1	1	1	NA
38	LD BHA	LD item (to pipe deck)	1	1	1	1	1	1	0
39	LD BHA	From derrick	1	1	1	1	1	1	0
40	LD BHA	Dump data / program	0	0	1	1	1	1	1
41	LD BHA	RA (remove)	0	0	1	1	1	1	1

Figure D.3: Full CSV-file (3/6).



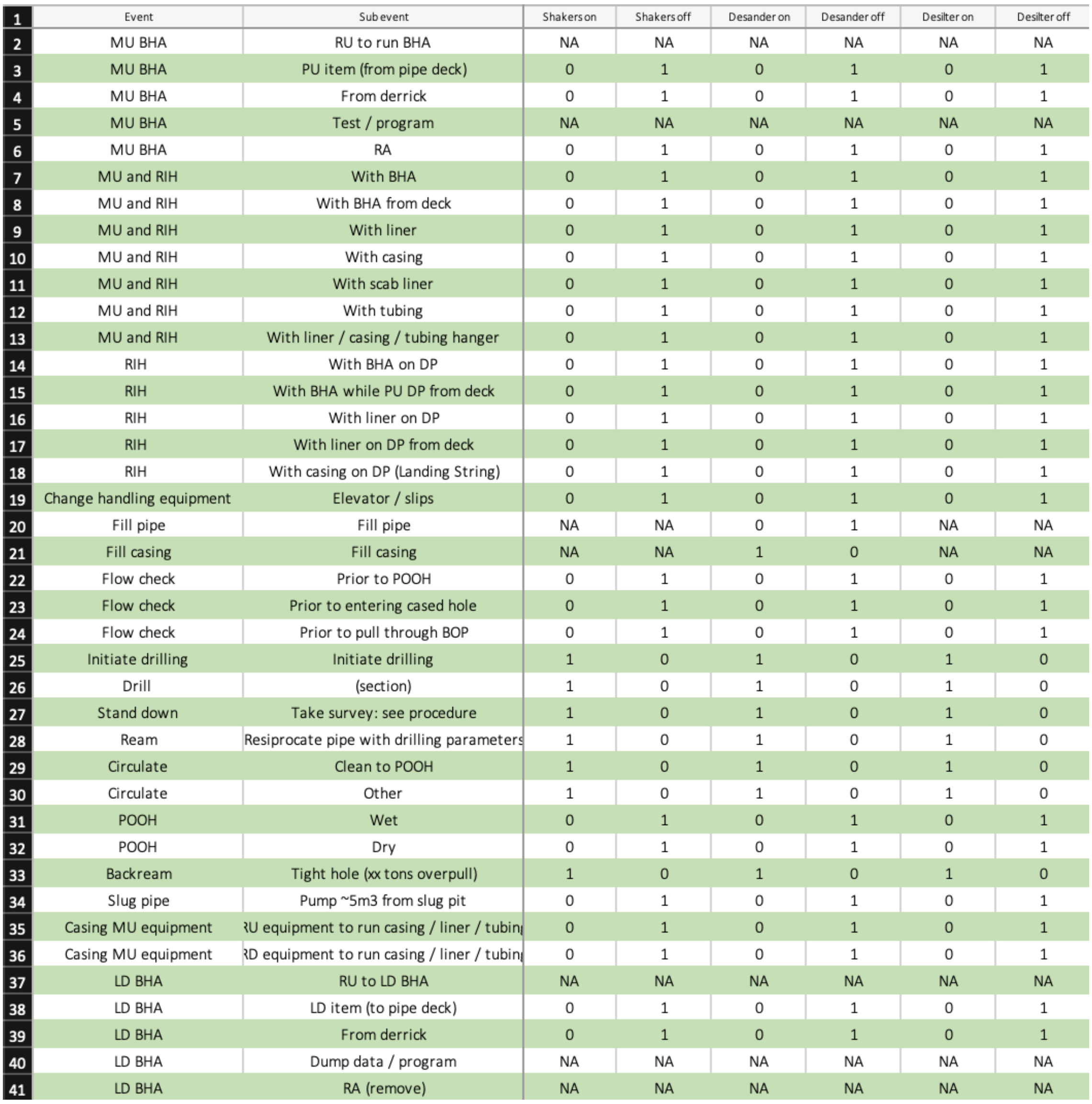
1	Event	Sub event	Rotation 1	Rotation 2	Rotation 3	Rotation 4	Stop rotation	Break circulation	Stage up to rate	Section / High flow
2	MU BHA	RU to run BHA	NA	NA	NA	NA	NA	NA	NA	NA
3	MU BHA	PU item (from pipe deck)	1	0	0	0	1	0	0	0
4	MU BHA	From derrick	1	1	1	1	1	0	0	0
5	MU BHA	Test / program	NA	NA	NA	NA	NA	1	1	0
6	MU BHA	RA	NA	NA	NA	NA	NA	1	1	0
7	MU and RIH	With BHA	NA	NA	NA	NA	NA	0	0	0
8	MU and RIH	With BHA from deck	NA	NA	NA	NA	NA	0	0	0
9	MU and RIH	With liner	NA	NA	NA	NA	NA	0	0	0
10	MU and RIH	With casing	NA	NA	NA	NA	NA	0	0	0
11	MU and RIH	With scab liner	NA	NA	NA	NA	NA	0	0	0
12	MU and RIH	With tubing	NA	NA	NA	NA	NA	0	0	0
13	MU and RIH	With liner / casing / tubing hanger	NA	NA	NA	NA	NA	0	0	0
14	RIH	With BHA on DP	1	0	0	0	1	0	0	0
15	RIH	With BHA while PU DP from deck	1	0	0	0	1	0	0	0
16	RIH	With liner on DP	1	0	0	0	1	0	0	0
17	RIH	With liner on DP from deck	1	0	0	0	1	0	0	0
18	RIH	With casing on DP (Landing String)	1	0	0	0	1	0	0	0
19	Change handling equipment	Elevator / slips	NA	NA	NA	NA	NA	NA	NA	0
20	Fill pipe	Fill pipe	NA	NA	NA	NA	NA	0	0	0
21	Fill casing	Fill casing	NA	NA	NA	NA	NA	0	0	0
22	Flow check	Prior to POOH	NA	NA	NA	NA	NA	0	0	0
23	Flow check	Prior to entering cased hole	NA	NA	NA	NA	NA	0	0	0
24	Flow check	Prior to pull through BOP	NA	NA	NA	NA	NA	0	0	0
25	Initiate drilling	Initiate drilling	0	1	0	0	0	1	1	1
26	Drill	(section)	1	1	1	1	1	1	1	1
27	Stand down	Take survey: see procedure	0	1	1	1	1	0	0	1
28	Ream	Resiprocate pipe with drilling parameters	1	1	1	1	1	0	0	1
29	Circulate	Clean to POOH	1	1	1	1	1	0	0	1
30	Circulate	Other	1	1	1	1	1	0	0	1
31	POOH	Wet	NA	NA	NA	NA	NA	0	0	0
32	POOH	Dry	NA	NA	NA	NA	NA	0	0	0
33	Backream	Tight hole (xx tons overpull)	1	1	1	1	1	1	1	1
34	Slug pipe	Pump ~5m3 from slug pit	1	1	1	1	1	1	0	0
35	Casing MU equipment	U equipment to run casing / liner / tubin	NA	NA	NA	NA	NA	0	0	0
36	Casing MU equipment	D equipment to run casing / liner / tubin	NA	NA	NA	NA	NA	0	0	0
37	LD BHA	RU to LD BHA	NA	NA	NA	NA	NA	NA	NA	NA
38	LD BHA	LD item (to pipe deck)	1	0	0	0	1	0	0	0
39	LD BHA	From derrick	1	0	0	0	0	0	0	0
40	LD BHA	Dump data / program	NA	NA	NA	NA	NA	0	0	0
41	LD BHA	RA (remove)	NA	NA	NA	NA	NA	0	0	0

Figure D.4: Full CSV-file (4/6).



1	Event	Sub event	ow (pressure test / f	stop pumping	no pumping	Trip tank open	Trip tank closed	Active pit open	Active pit closed
2	MU BHA	RU to run BHA	NA	NA	NA	1	0	0	1
3	MU BHA	PU item (from pipe deck)	0	0	1	1	0	0	1
4	MU BHA	From derrick	0	0	1	1	0	0	1
5	MU BHA	Test / program	1	1	0	0	1	1	0
6	MU BHA	RA	1	1	1	1	0	0	1
7	MU and RIH	With BHA	0	0	1	1	0	0	1
8	MU and RIH	With BHA from deck	0	0	1	1	0	0	1
9	MU and RIH	With liner	0	0	1	1	0	0	1
10	MU and RIH	With casing	0	0	1	1	0	0	1
11	MU and RIH	With scab liner	0	0	1	1	0	0	1
12	MU and RIH	With tubing	0	0	1	1	0	0	1
13	MU and RIH	With liner / casing / tubing hanger	0	0	1	1	0	0	1
14	RIH	With BHA on DP	0	0	1	1	0	0	1
15	RIH	With BHA while PU DP from deck	0	0	1	1	0	0	1
16	RIH	With liner on DP	0	0	1	1	0	0	1
17	RIH	With liner on DP from deck	0	0	1	1	0	0	1
18	RIH	With casing on DP (Landing String)	0	0	1	1	0	0	1
19	Change handling equipment	Elevator / slips	NA	NA	1	1	0	0	1
20	Fill pipe	Fill pipe	1	1	NA	1	0	0	1
21	Fill casing	Fill casing	1	1	NA	2	1	1	0
22	Flow check	Prior to POOH	0	0	1	1	0	0	1
23	Flow check	Prior to entering cased hole	0	0	1	1	0	1	0
24	Flow check	Prior to pull through BOP	0	0	1	1	0	2	1
25	Initiate drilling	Initiate drilling	0	0	0	0	1	1	0
26	Drill	(section)	0	1	0	0	1	1	0
27	Stand down	Take survey: see procedure	0	1	0	0	1	1	0
28	Ream	Resiprocate pipe with drilling parameters	0	1	0	0	1	1	0
29	Circulate	Clean to POOH	0	1	0	0	1	1	0
30	Circulate	Other	0	1	0	0	1	2	1
31	POOH	Wet	0	0	1	1	0	0	1
32	POOH	Dry	0	0	1	1	0	1	0
33	Backream	Tight hole (xx tons overpull)	0	1	0	0	1	1	0
34	Slug pipe	Pump ~5m3 from slug pit	1	1	0	1	0	0	1
35	Casing MU equipment	RU equipment to run casing / liner / tubin	0	0	1	1	0	0	1
36	Casing MU equipment	RD equipment to run casing / liner / tubin	0	0	1	1	0	0	1
37	LD BHA	RU to LD BHA	NA	NA	NA	1	0	0	1
38	LD BHA	LD item (to pipe deck)	0	0	1	1	0	0	1
39	LD BHA	From derrick	0	0	1	1	0	0	1
40	LD BHA	Dump data / program	0	0	1	1	0	0	1
41	LD BHA	RA (remove)	0	0	1	1	0	0	1

Figure D.5: Full CSV-file (5/6).



1	Event	Sub event	Shakers on	Shakers off	Desander on	Desander off	Desilter on	Desilter off
2	MU BHA	RU to run BHA	NA	NA	NA	NA	NA	NA
3	MU BHA	PU item (from pipe deck)	0	1	0	1	0	1
4	MU BHA	From derrick	0	1	0	1	0	1
5	MU BHA	Test / program	NA	NA	NA	NA	NA	NA
6	MU BHA	RA	0	1	0	1	0	1
7	MU and RIH	With BHA	0	1	0	1	0	1
8	MU and RIH	With BHA from deck	0	1	0	1	0	1
9	MU and RIH	With liner	0	1	0	1	0	1
10	MU and RIH	With casing	0	1	0	1	0	1
11	MU and RIH	With scab liner	0	1	0	1	0	1
12	MU and RIH	With tubing	0	1	0	1	0	1
13	MU and RIH	With liner / casing / tubing hanger	0	1	0	1	0	1
14	RIH	With BHA on DP	0	1	0	1	0	1
15	RIH	With BHA while PU DP from deck	0	1	0	1	0	1
16	RIH	With liner on DP	0	1	0	1	0	1
17	RIH	With liner on DP from deck	0	1	0	1	0	1
18	RIH	With casing on DP (Landing String)	0	1	0	1	0	1
19	Change handling equipment	Elevator / slips	0	1	0	1	0	1
20	Fill pipe	Fill pipe	NA	NA	0	1	NA	NA
21	Fill casing	Fill casing	NA	NA	1	0	NA	NA
22	Flow check	Prior to POOH	0	1	0	1	0	1
23	Flow check	Prior to entering cased hole	0	1	0	1	0	1
24	Flow check	Prior to pull through BOP	0	1	0	1	0	1
25	Initiate drilling	Initiate drilling	1	0	1	0	1	0
26	Drill	(section)	1	0	1	0	1	0
27	Stand down	Take survey: see procedure	1	0	1	0	1	0
28	Ream	Resiprocate pipe with drilling parameters	1	0	1	0	1	0
29	Circulate	Clean to POOH	1	0	1	0	1	0
30	Circulate	Other	1	0	1	0	1	0
31	POOH	Wet	0	1	0	1	0	1
32	POOH	Dry	0	1	0	1	0	1
33	Backream	Tight hole (xx tons overpull)	1	0	1	0	1	0
34	Slug pipe	Pump ~5m3 from slug pit	0	1	0	1	0	1
35	Casing MU equipment	RU equipment to run casing / liner / tubin	0	1	0	1	0	1
36	Casing MU equipment	RD equipment to run casing / liner / tubin	0	1	0	1	0	1
37	LD BHA	RU to LD BHA	NA	NA	NA	NA	NA	NA
38	LD BHA	LD item (to pipe deck)	0	1	0	1	0	1
39	LD BHA	From derrick	0	1	0	1	0	1
40	LD BHA	Dump data / program	NA	NA	NA	NA	NA	NA
41	LD BHA	RA (remove)	NA	NA	NA	NA	NA	NA

Figure D.6: Full CSV-file (6/6).

E DOP for drilling 17,5” section

See next page.

Distributed to:
Drilling Supervisor, 6 x Drilling Contractor, Cementer, Mud engineer, Mud logger, DD, MWD.

NTNU Driller MODU

Detailed Operation Procedure
DOP #10

Procedure:

Drill 17 ½” hole

Prepared by:
Onshore Operation Planner

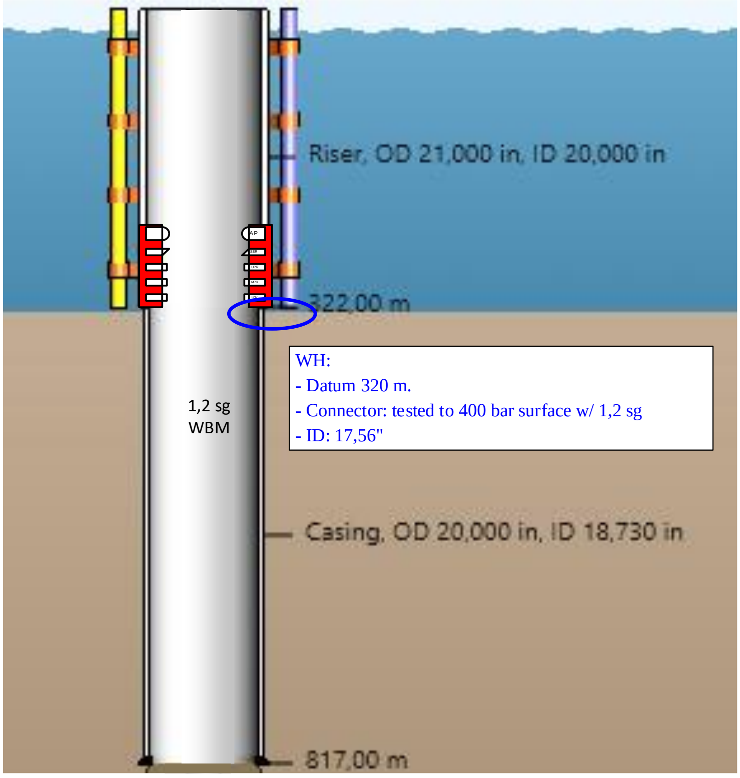
Checked by:
Offshore Operation Planner: ___________________________

Approved by:
Drilling Contractor Tool pusher: ___________________________

Drilling Supervisor: ________________________________

Well Status:

- The well is filled with 1,2 sg. WBM.
- The BOP is installed. The WH connector is pressure tested to 400 bar surface with 1,2 sg. WBM.
- WH Datum @: 320 m MD.
- 26" hole is drilled to 820 m MD
- TOC 20" casing at 322 m MD
- The well is tested to 130 bar with 1,2 sg WBM.
- 18 ¾" Nominal Bore Protector installed. (ID 17,595")
- WH ID: 17,562"



(Bjorbre)

Objectives:

- Provide integrity for drilling the next section: Case off the weak Formation L and Formation B. If 13 ⅜" shoe is set minimum 50 m MD into the Formation S, there is space for a potential whipstock and sidetrack while still inside this formation.
- Drill out 20" shoe track and clean rat hole + 3 m new formation
- Perform FIT to 1,7 sg EMW.
- Drill 17 ½" hole to 2182 m MD.

Risks:

- Shock related failure of bit and drilling BHA when drilling stringers and shoe track.
- Hard stringers leading to wash out
- Unstable hole, problematic hole cleaning, tight spots, stuck BHA. Focus on correct mud properties.

- Weak and unstable Formation B and Formation L require careful mud weight up schedule to avoid losses.
- Losses in H group sands if increasing mud weight too early.
- Risk of high gas readings drilling through Formation B/Formation L/Formation S.

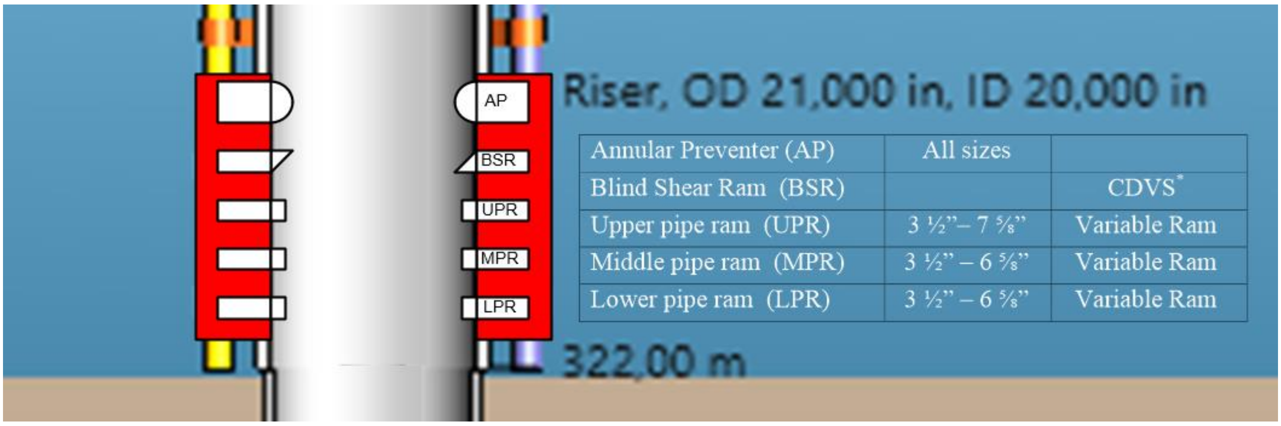
Safety:

- Perform a pre job meeting with all relevant personnel on drill floor right before the operation starts.
- Focus on red zone.
- Post job meeting shall be performed for experience transfer and operation optimization, where the key issue is to involve personnel performing the operations.
- All operations that are defined as a lifting operation need to be performed according to regulations with regards to certification, documentation and procedures.

BOP configuration:

Name (Abreviation)	Size	Type
Annular Preventer (AP)	All sizes	
Blind Shear Ram (BSR)		CDVS*
Upper pipe ram (UPR)	3 ½"– 7 ⅝"	Variable Ram
Middle pipe ram (MPR)	3 ½" – 6 ⅝"	Variable Ram
Lower pipe ram (LPR)	3 ½" – 6 ⅝"	Variable Ram

* CDVS (Cable double-V shear rams: link)



Riser, OD 21,000 in, ID 20,000 in

Annular Preventer (AP)	All sizes	
Blind Shear Ram (BSR)		CDVS*
Upper pipe ram (UPR)	3 ½"– 7 ⅝"	Variable Ram
Middle pipe ram (MPR)	3 ½" – 6 ⅝"	Variable Ram
Lower pipe ram (LPR)	3 ½" – 6 ⅝"	Variable Ram

322,00 m

(Bjorbre)

Space-out requirements:

Space out to avoid making connection with bit at critical depths such as in BOP, 20" casing shoe/rat hole.

Critical depths:

- RKB – Top BOP: 314 m MD
- RKB – Annular: 315 m MD

- RKB – WH datum: 320 m MD
- RKB – Landing collar : 805 m MD
- RKB – 20” Casing shoe: 817 m MD
- RKB – TD 26” hole: 820 m MD

String configuration:

String:	Connection:	Make up nominal: Nm	Tensile strength: metric ton
5 ½” S-135	VX-57	65.000	302
5 ½" HWDP	VT-57	72.000	474
8” DC	6 5/8” Reg	72.000	765

Reference documents to this procedure:

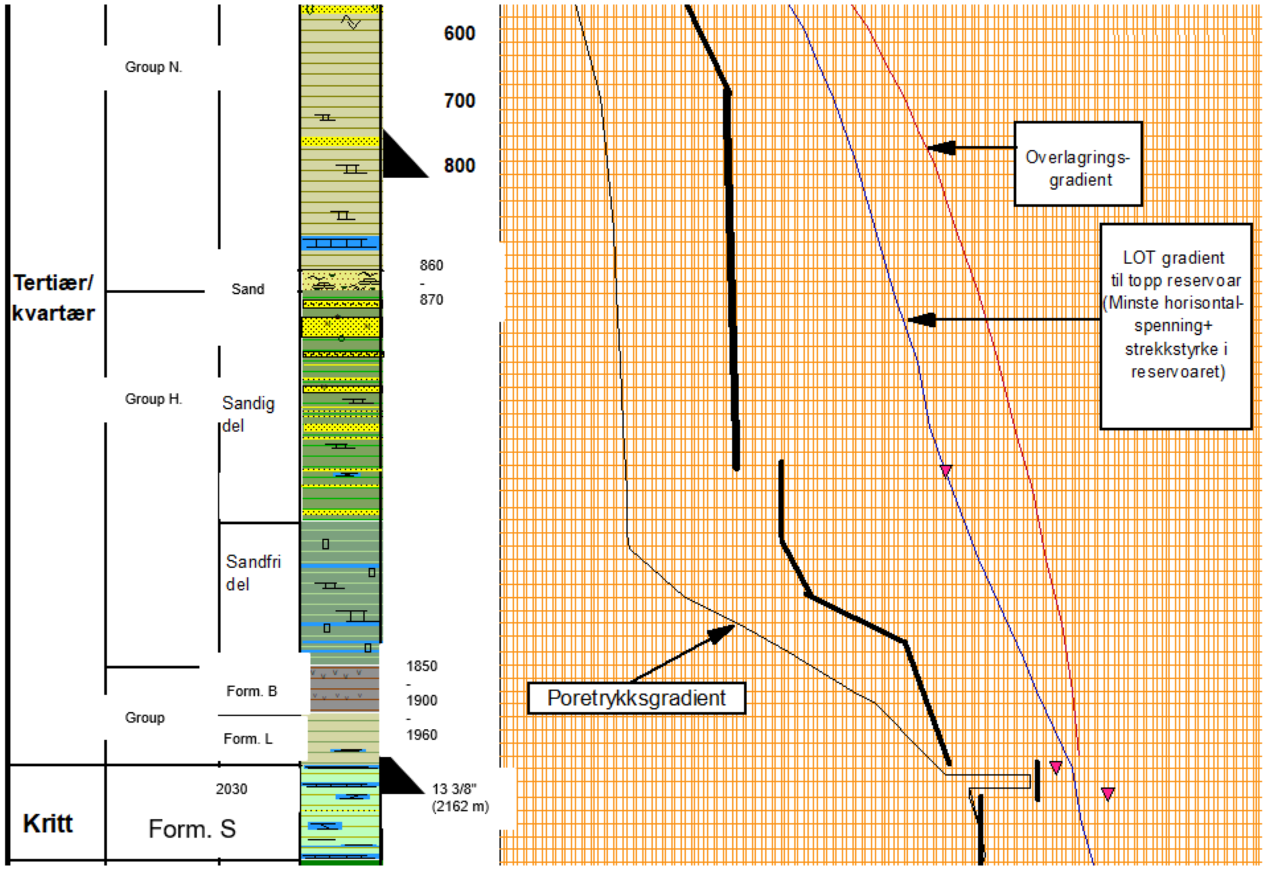
1. Individual Activity Program for Drilling “current well”.
2. Project change log for Drilling “current well”
3. Section risk matrix and Action log.
4. Bottom hole assembly, with running procedures from Directional Drilling Supplier.
5. Drilling fluid program and manual.
7. Drilling contractors detailed procedures.

Abbreviations

CDVS - Cable double-V shear rams
DD – Directional Driller
LCM – Lost Circulation Material
MWD – Measurement While Drilling (tool and role)
NBP – Nomonal Bore Protector
SCR – Slow Circulating Rates
WBM – Water based Mud
WH – Well Head
XO - Crossover
XT – Christmas Tree

Geological prognosis “current well”

Stratigraphic Tops	Prognosed Depth (RKB 22m TVD)	Prognosed Depth [m MD RKB]	Seismic Uncertainty [TVD]
Top Sand (Group N.)	860	870	±20
Top H Group	922	933	±20
Base ‘Sandy part of group H	1640	1686	±20
Top formation B	1850	1900	±20
Top formation L	1900	1960	±20
Top form S.	2030	2100	±20



(Bjorbre)

Pre-section preparations checklist – All companies involved:

#	CHECKLIST	RESPONSIBLE	VERIFIED
1.	Pre job meeting carried out.	Drilling Sup	
2.	Risk Analysis reviewed, implemented and signed out	Drilling Sup	
3.	Correct crossovers for planned equipment in place. Including documentation.	Tool pusher	
4.	5 ⅞" DP with XT-57 connections for drilling section.	Tool pusher	
5.	Checklist for loose lifting equipment to be signed out	Tool pusher	
6.	BHA sheets to be signed by Toolpusher prior to run in hole.	Tool pusher	
7.	5 ½" liners installed in all (four) mud pumps, and correct rupture disc.	Tool pusher	
8.	Hang-off stand correctly built dressed for 18 ¾" NBP and racked back in Derrick.	Ass Driller / Subsea comp.	
9.	Hours on wash pipe and saver sub checked prior to RIH.	Ass Driller	
10.	Magnets installed in flow divider and flow line. Plan for cleaning, inspection and record of metal in place.	Ass. Driller	
11.	XO's to be used in BHA shall be measured, drifted and checked for valid certificates.	Ass. Driller.	
12.	All down hole tools inspected and measured. Verify that pre made subs have correct torque (check documentation).	Ass. Driller/DD	
13.	Verify that pressure on high-pressure discharge/low pressure pulsation dampers are according to recommendations given in NOV manual.	Derrick man	
14.	Check magnets every shift to be included in Mudloggers midnight report.	Mudlogger	
15.	Suficient mud (new and old) and chemicals for drilling the section.	Mud engineer	
16.	Ensure chemicals or excess mud are on board for making minimum one hole volume of 1,20 sg WBM (contingency if major losses to formation.).	Mud engineer	
17.	LCM materials ready in case of losses if drilling in loss zone.	Mud engineer	
18.	LCM (350 kg/m3) pill ready for FIT	Mud engineer	
19.	Ensure enough optimized screens for drilling on board.	Mud engineer	
20.	Mud engineer to check and calibate mud balance on shaker and pump room every shift.	Mud engineer	
21.	DD, MWD checklists completed. And provide fishing diagram.	DD / MWD	
22.	Non ported float installed in BHA.	DD	
23.	Jar sheet to be prepared on DF.	Ass. Driller.	
24.	Procedure for jar available on drill floor.	Service provider	
25.	DD to check all BU bits, jets, and flow restrictor kit	DD	
26.	Roadmaps for section available on rig floor.	Geologist/ Mudlogger	
27.	Check Drift sub and Drift.	Ass Driller	
28.	Perform kick drill and pit drill during the operation	Toolpusher	

Operator representative: ______________________________

Detailed activities: “Drill 17 ½” hole”

#	ACTIVITY DESCRIPTION	PARALLEL ACTIVITIES / ITEMS TO REMEMBER
Pre-Job meeting		
1.	• Hold pre-job meeting with involved personnel prior to M/U of BHA	**<u>Drilling Contractors procedures for upcoming operations:</u>** • Make up 17 ½” BHA • Drill 17 ½” section
Make Up BHA and RIH		
2.	• Make up 17 ½” BHA according to DD instructions.	• Several components already racked back from 26” section
3.	• RIH with BHA on 5 ⅞” DP. Compensate through BOP / WH area (310 – 325 m).	
4.	• Continue RIH with BHA on 5 ⅞” DP to ~ 10 m above landing collar inside the 20” casing. • Fill pipe with 1,19 sg WBM and take up- and down weights @ ~ 795 meters. ○ Up weight: __________ton. ○ Down Weight: ________ton. ○ Shallow test MWD prior to start drilling cement.	▪ Landing collar @: 805 m. MD ▪ Install correct screens on shakers while running in hole.
Perform Choke drill		
5.	• Perform choke drill with crew, max 30 Bar on choke.	

#	ACTIVITY DESCRIPTION	PARALLEL ACTIVITIES / ITEMS TO REMEMBER
Drill out shoe track and rat hole		
6.	• Drill out float/cement/shoe track from 805 m to 816 m MD. (That will leave 1 meter shoe track left to drill.) **Drilling Parameters Cement** ➢ WOB: Start with 1-2 tons and gradually increase to 5-10 tons. ➢ Rotation 40-60 rpm or according to Directional Driller instructions. ➢ Flow 4100 lpm. ➢ Flow range 4000 – 4250 lpm. • Circulate hole clean for cement contamination before displacing to 1.20 Sg drill mud.	• Cement and cuttings to go from shakers to overboard line. • Driller and Mudlogger to reset stroke counters on mud pumps when start drilling. • Magnets installed in flow line. • If possible space out to start drilling shoe on middle of a stand. • Limit torque setting on Top Drive to a minimum when drilling out cement. • Keep good record of drilling trends.
Displace to 1,20 sg drill mud and take choke line friction.		
7.	• Hold pre-job meeting with involved personnel prior to displace to1,20 sg Glydrill WBM.	• Empty trip tank prior to displace. • Displace booster line theoretically. • Kill/Choke line to be U-tube after displacing.
8.	• Displace to 1,20 sg WBM, according to mud engineers separate plan. • Fill up trip tank when mud in return. • Circulate until even mud weights in/out.	• Mud engineer to be on shakers and decide when mud in return is OK. • Mud engineer to optimize shakers for start drilling formation.

#	ACTIVITY DESCRIPTION	PARALLEL ACTIVITIES / ITEMS TO REMEMBER
9.	• Take SCR at 20, 30, and 40 SPM up riser and kill, choke-line.	• Take torque/drag readings and measure up/down weights prior to start drilling. ○ Up weight : _______________ton ○ Down weight : _____________ton ○ Torque 40 RPM :_____________kNm
Clean rat hole and drill 3 m new formation and prepare for FIT		
10.	• Continue drill out rest of shoe track/shoe, 1 meter from 816 m MD to 817 m MD. • Drill rat hole carefully in steps of 1 meter. Drilling Parameters rat hole ▪ WOB: Start with **2-5** tons ▪ Rotation **60 - 80** rpm ▪ Flow **4250** lpm. Take caution when cleaning out rat-hole to avoid pack-off, stuck pipe and high shocks/ vibrations. Drill **1** m and pull back inside casing shoe to ensure free passage upwards.	• 20” Shoe at 817 m MD • Rat hole is approximately 5 m long. • Cementer to pressure test unit/Surface lines to 100 bar prior to FIT. ▪ **Stop rotation and decrease pump rate when pulling bit into casing shoe**, DD had to be present at drill floor when pulling bit into casing shoe.
11.	• Spot 7 m³ LCM (350 kg/m3) pill in open hole and 50 m inside 20” casing prior to FIT. • Pull bit inside 20” shoe (above top LCM pill). • Circulate 1 well volume @ 4250 lpm. • Space out and close annular.	• Stop rotation when pulling bit into casing. • Pump LCM according to MWD instruction, below MWD turn on rate (max 1000 lpm). • After FIT, pump up FIT survey data from MWD as pr. MWD operator’s instruction.

#	ACTIVITY DESCRIPTION	PARALLEL ACTIVITIES / ITEMS TO REMEMBER
12.	• Pressurize LCM pill to 1,57 sg EMW (29,7 bar) by using 1,20 sg WBM down string and kill line with cement unit. • Pressure up to 10 bar. Bleed off to 5-7 bar reference pressure. • Zero stroke counter. • Pressure up to 29,7 bar with constant flow rate 100 l/min. • Hold pressure for 15 min. • Bleed off pressure slowly. • Record volumes pumped and bled back. • Open annular	• If pressure drops (5-10 bar), re-pressurize to FIT pressure. • Volume to pressure up: 81,53 m3 • Expected test volume: 115 liter. • Actual test volume: _________ liter
13.	• Start circulation and wash down to TD. • Drill 3 m of new formation. Drilling Parameters formation ○ **WOB: Start with** 2-5 **tons** ○ **Rotation** 60 - 80 **rpm** ○ **Flow** 4500 **lpm. (**4400-5000**)** • Circulate well clean and condition mud to even MW in/out. • Pull string into 20” shoe. • Space out and close annular.	
Perform FIT test		
14.	• Perform FIT to 1,57 sg EMW by using 1,20 sg WBM down string and kill line with cement unit. • Pressure up to 10 bar. Bleed off to 5 - 7 bar reference pressure. • Re-zero volume counter. • Pressure up to 29,7 bar with constant flow rate of 100 l/min.	• Operator representative to be in cement unit during the operation. • Volume to pressure up: 81,83 m3 • Expected test volume: 116 liter. • Actual test volume: _________ liter • Evaluate FIT test results.

<table>
<tr><th>#</th><th>ACTIVITY DESCRIPTION</th><th>PARALLEL ACTIVITIES / ITEMS TO REMEMBER</th></tr>
<tr><td></td><td>**FIT continued:**

• Hold pressure for 10-15 min.
• Bleed off pressure slowly to ref. pressure.
• Record volumes pumped and bled back.
• Bleed off remaining pressure.
• Open annular</td><td></td></tr>
<tr><td colspan="3">**Drill 17 ½” hole**</td></tr>
<tr><td>**15.**</td><td>• Drill 17 ½” hole to approximately 2182 m MD / 2162 m TVD.
• MW to be increased in steps to 1,45 sg according to table below. To be followed up of the mud engineer.

<table>
<tr><th>Action</th><th>Approx. MD</th><th>MW</th><th>Min KCl</th></tr>
<tr><td>Incr. MW</td><td>1400</td><td>1,20</td><td>140</td></tr>
<tr><td>Incr. MW</td><td>1550</td><td>1,35</td><td>140</td></tr>
<tr><td>Max/planned MW</td><td>1775</td><td>1,45</td><td>140 - 150</td></tr>
<tr><td>TD</td><td>TD</td><td>1,45</td><td>140 - 150</td></tr>
</table>

• Perform survey cluster shot 50 m below shoe.

Drilling Parameters with 1,38 sg mud:
• WOB 5 –20 tons.
• RPM 80-160 rpm / according to DD’s instructions.
• Plan flow 4800 lpm (range 4300 – 4900 lpm).</td><td>• Max ROP: Dictated by hole cleaning and cuttings handling on surface.
• Stringers to be reamed from above.
• Perform survey every std. (minimum every 100 m)
• Be aware of possible plugging of shaker screens when drilling through sand.
• Mudlogger to have focus on gas equipment and background gas while drilling
• Abnormal gas peaks in Formation B and L may occur due to gas chimney.
• Actions when observing gas peaks:
  ✓ Flow check and circulate out the gas.
  ✓ If several connections gas occurs, evaluate to increase MW with 3-4 points.

• 1664 m TVD hard stinger expected (approx. 1710 m MD)
• 1650 – 1670 m TVD may be a loss zone.</td></tr>
</table>

#	ACTIVITY DESCRIPTION	PARALLEL ACTIVITIES / ITEMS TO REMEMBER
Circulate hole clean		
16.	• At TD Circulate minimum 2 X B/U with max flow 4900 lpm and 80 - 160 RPM or until hole is clean. **<u>Perform readings prior to POOH:</u>** ▪ Free rotation 30 RPM:_______kNm ▪ Up Weight: ________ Ton. ▪ Down Weight: _______ Ton. • Record SCRs at 20, 30 and 40 SPM prior to POOH	• Reciprocate string during circulation. Don’t stop on same spot during circulation. • • Geologist to provide trip risk log. <u>SCR readings:</u> ▪ 20 SPM: _________ ▪ 30 SPM: _________ ▪ 40 SPM: _________
17.	• Take 10 min flow check. ▪ Have low rotation (5 – 10 rpm) on string during flow check.	• Updated surge &swab sheet to be made.
18.	• Pull 5 std wet, and if hole is ok pump slug.	• Install plate and wiper on top of diverter.
POOH.		
19.	POOH from TD to 20” shoe at 817 m MD: • Take check survey at same depth as Cluster shot. • Take care when pulling BHA and bit into shoe track. • Take 10 min flow check.	• Limit tripping speed according to surge/swab calculations. • Compensate string.
20.	• Continue POOH until top of BHA is below BOP. • • Flow check well 10 min. • Continue to POOH.	▪ Perform pre-job meeting w/crew before handling BHA. ▪ Remove plate and wiper prior to pull BHA. ▪ Note that Remote operated slips cannot be used with string weight less than 10 tons (manual slips only)

#	ACTIVITY DESCRIPTION	PARALLEL ACTIVITIES / ITEMS TO REMEMBER
21.	• Rack BHA in derrick. ▪ DD an MWD to check and inspect Bit and other vital BHA components prior to racking BHA. ▪ DD and MWD to discuss with driller / ass. Driller plans for reusing components in 17 ½” BHA in 12 ¼” drilling BHA.	• Minimize tme before running casing (hole stability). • ▪ DD to note: ○ Bit gauge: ________ inch. ○ Bit grade: ________________. ○ Any balling on bit/BHA.
Hand in procedure with comments to operator representative (drilling engineer).		
Next operation: Pull 18 ¾” NBP.		

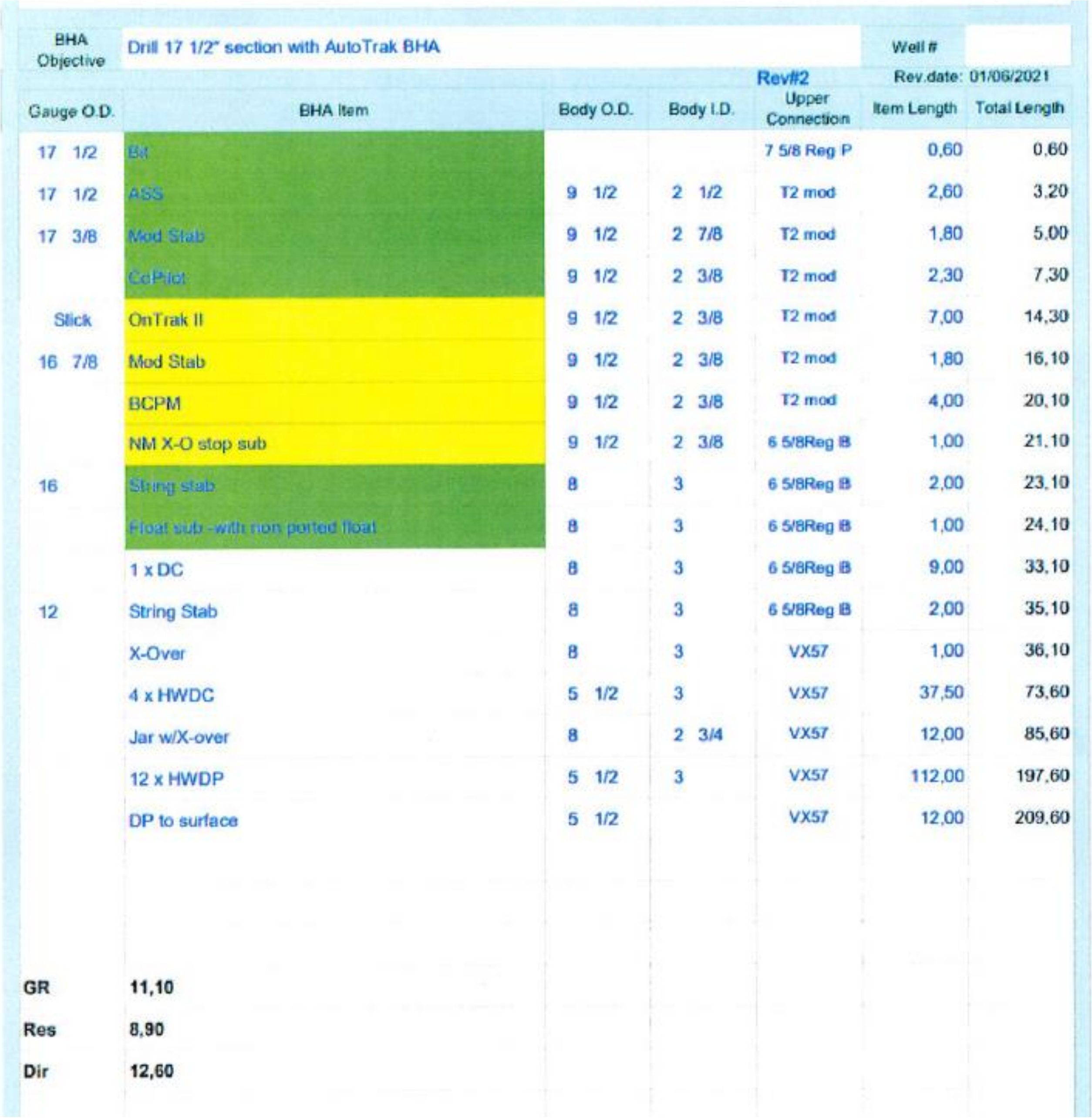
Attachments:

- Risk Assessment
- BHA
- WBS
- Drilling-Connection and survey recommendations

Risk assesment:

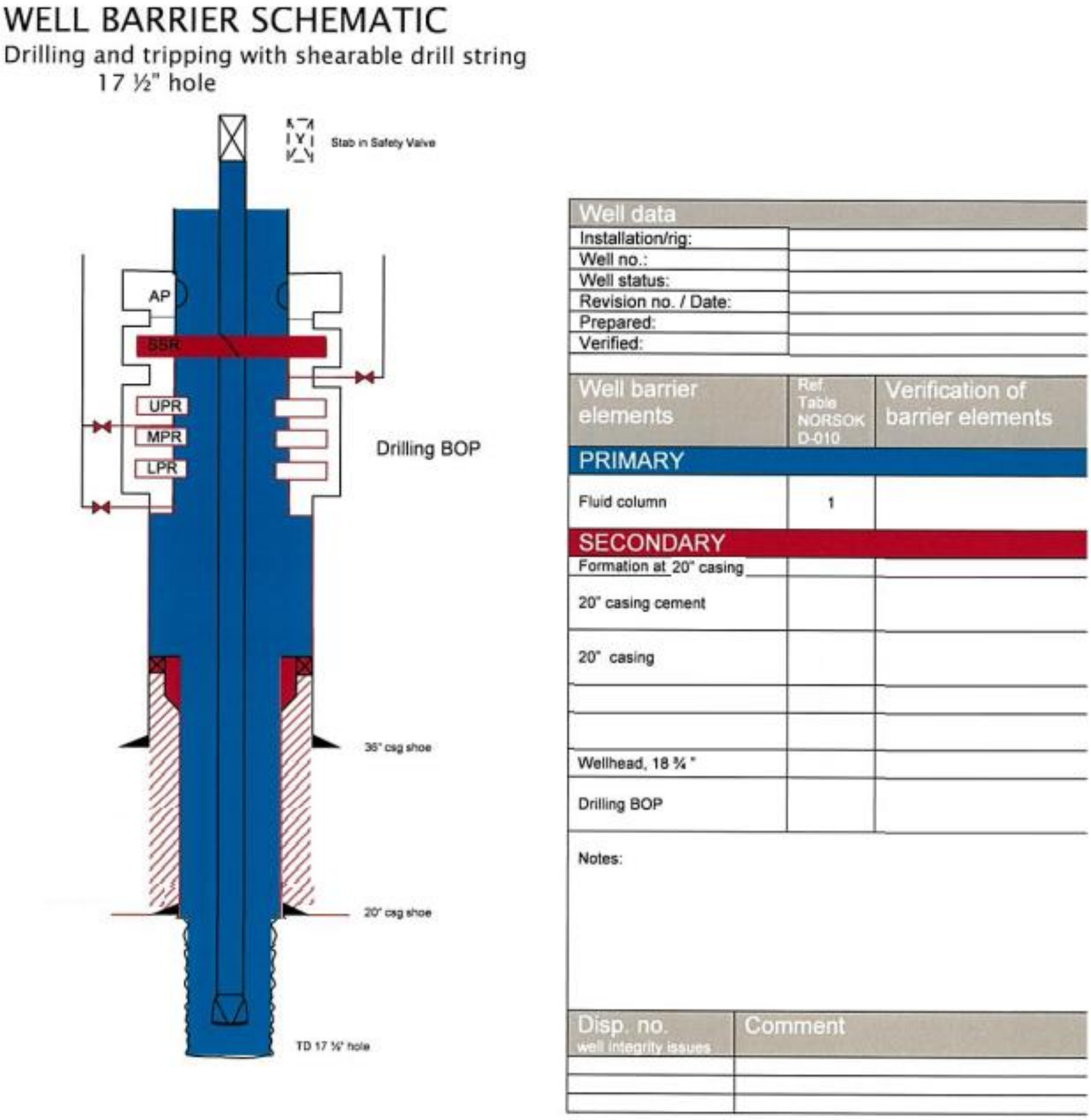
ID	Hazard/Opportunities	Risk Description		Existing safeguards	Risk before mitigation				Risk mitigation				Risk after mitigation				Risk status	Cov. by ref. data	Comment
		Causes	Consequences		Prob	Impact			Measures	Status	Responsible	Deadline	Prob	Impact					
						HSE	OBJ	TC						HSE	OBJ	TC			
									Drilling of [redacted] 17 1/2"section / running intermediate 13 3/8" casing										
R6-1	Drilling problems, ref Well 1	Geology related drilling problems, ledges	Sidetrack, further problems	Wellpath considerations are already optimised for this section with lowest possible inclination, and tangent through section. No problems similar to Well 1	P2	I1	I4	I4	Plan for using experinence gained on O-1 an P-1.	Proposed	Drilling Eng.	[redacted]	P1	I1	I4	I4			
									Order Kymera bit for 14 3/4	Proposed	Drilling Eng.								
									Best practice for drilling stringers										
R6-6	Hole problems due to Unstable Formation B and L	Unstable Formation B and L	• Mud losses. • Hole collapse • Stuck BHA	Inclination less than 45 deg based on experience from Well 1	P2	I2	I2	I2	- Optimize mud parameters. Evaluate to add LCM/KCL in active system prior to drilling through Formation B .-Use LEDO	Verified	Drilling		P2	I2	I2	I2			

BHA



BHA Objective	Drill 17 1/2" section with AutoTrak BHA				Well #	
				Rev#2	Rev.date:	01/06/2021
Gauge O.D.	BHA Item	Body O.D.	Body I.D.	Upper Connection	Item Length	Total Length
17 1/2	Bit			7 5/8 Reg P	0,60	0,60
17 1/2	ASS	9 1/2	2 1/2	T2 mod	2,60	3,20
17 3/8	Mod Stab	9 1/2	2 7/8	T2 mod	1,80	5,00
	CoPilot	9 1/2	2 3/8	T2 mod	2,30	7,30
Slick	OnTrak II	9 1/2	2 3/8	T2 mod	7,00	14,30
16 7/8	Mod Stab	9 1/2	2 3/8	T2 mod	1,80	16,10
	BCPM	9 1/2	2 3/8	T2 mod	4,00	20,10
	NM X-O stop sub	9 1/2	2 3/8	6 5/8Reg B	1,00	21,10
16	String stab	8	3	6 5/8Reg B	2,00	23,10
	Float sub -with non ported float	8	3	6 5/8Reg B	1,00	24,10
	1 x DC	8	3	6 5/8Reg B	9,00	33,10
12	String Stab	8	3	6 5/8Reg B	2,00	35,10
	X-Over	8	3	VX57	1,00	36,10
	4 x HWDC	5 1/2	3	VX57	37,50	73,60
	Jar w/X-over	8	2 3/4	VX57	12,00	85,60
	12 x HWDP	5 1/2	3	VX57	112,00	197,60
	DP to surface	5 1/2		VX57	12,00	209,60
GR	**11,10**					
Res	**8,90**					
Dir	**12,60**					

WBS



WELL BARRIER SCHEMATIC

Drilling and tripping with shearable drill string
17 ½" hole

Well data	
Installation/rig:	
Well no.:	
Well status:	
Revision no. / Date:	
Prepared:	
Verified:	

Well barrier elements	Ref. Table NORSOK D-010	Verification of barrier elements
PRIMARY		
Fluid column	1	
SECONDARY		
Formation at 20" casing		
20" casing cement		
20" casing		
Wellhead, 18 ¾ "		
Drilling BOP		

Notes:

Disp. no. well integrity issues	Comment

Drilling-Connection and survey recommendations

RSS Connection Procedure

Important! Survey after connect ion

1. Drill stand down (Driller to inform on radio when connection passes table for depth calibration - do not stop)
2. Pull off bottom + 2m using 1 minute
    - Keep drilling RPM
    - Keep 100% flow
3. Stop rotation
4. P/U and record stable up and down weight (3 - 5m)
    - Speed up and down, max 10 m/min
    - 100% flow moving up (If ECD is high, stop 10sec on top before moving down)
    - 80% flow moving down (100% flow also down if conditions allows)
5. Set slips and make connection
    - Start filling pipe immediately w/ 400 1pm while making up connection
    - Do not start rotation
6. Survey and Start up sequence
    - Increase to 80 % flow (1-2 min)
    - Open compensator if required
    - Do not move pipe (up/down)
    - When survey taken downhole, MWD engineer to inform driller immediately
    - Increase to 80 RPM (1/2 min), hold (1/2 - 1 min) while reading free rotating torque, keep 80% flow
    - Increase to drilling RPM, keep 80% flow, MWD to confirm survey received
    - Increase to 100% flow
7. Drill ahead
    - Check pressure response when going on bottom

Magnus Bareksten & Lars Djuve Vambheim

Digital Well Planning

NTNU
Norwegian University of
Science and Technology