



**FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE**

ASSIGNMENT OF MASTER'S THESIS

Title: A Case Study and Proof of Concept of the Application of Machine Learning to Polarion's ALM Software
Student: Bc. Michal Sláma
Supervisor: Ing. Jurij Černikov
Study Programme: Informatics
Study Branch: Web and Software Engineering
Department: Department of Software Engineering
Validity: Until the end of winter semester 2019/20

Instructions

The aim of this thesis is to analyze and identify machine learning (ML) use cases that would prove valuable for Polarion's application lifecycle management (ALM) software. A proof of concept prototype will be supplied for the selected use case.

1. Analyze and describe Polarion in order to identify suitable use cases to apply ML to.
2. Provide a review of ML frameworks and algorithms that are relevant for such an application.
3. Describe several use cases for ML and define their benefit to both Polarion as a business and the users that deploy it.
4. Choose a scenario from the previous investigation and implement a proof of concept prototype.
5. Discuss the possibility of the full implementation and deployment of the previous prototype into the production environment.

References

Will be provided by the supervisor.

Ing. Michal Valenta, Ph.D.
Head of Department

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
Dean

Prague March 1, 2018



**FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE**

Master's thesis

A Case Study and Proof of Concept of the Application of Machine Learning to Polarion's ALM Software

Bc. Michal Sláma

Katedra softwarového inženýrství
Supervisor: Ing. Jurij Černíkov

May 22, 2018

Acknowledgements

I would like to thank to my supervisor for his extraordinary leadership and valuable advice during the whole process of writing this thesis.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended. In accordance with Article 46(6) of the Act, I hereby grant a nonexclusive authorization (license) to utilize this thesis, including any and all computer programs incorporated therein or attached thereto and all corresponding documentation (hereinafter collectively referred to as the “Work”), to any and all persons that wish to utilize the Work. Such persons are entitled to use the Work in any way (including for-profit purposes) that does not detract from its value. This authorization is not limited in terms of time, location and quantity. However, all persons that makes use of the above license shall be obliged to grant a license at least in the same scope as defined above with respect to each and every work that is created (wholly or in part) based on the Work, by modifying the Work, by combining the Work with another work, by including the Work in a collection of works or by adapting the Work (including translation), and at the same time make available the source code of such work at least in a way and scope that are comparable to the way and scope in which the source code of the Work is made available.

In In Prague on May 22, 2018

.....

Czech Technical University in Prague

Faculty of Information Technology

© 2018 Michal Sláma. All rights reserved.

This thesis is school work as defined by Copyright Act of the Czech Republic. It has been submitted at Czech Technical University in Prague, Faculty of Information Technology. The thesis is protected by the Copyright Act and its usage without author's permission is prohibited (with exceptions defined by the Copyright Act).

Citation of this thesis

Sláma, Michal. *A Case Study and Proof of Concept of the Application of Machine Learning to Polarion's ALM Software*. Master's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2018.

Abstrakt

Diplomová práce obsahuje přehled možných použití strojového učení v aplikaci Polarion ALM. Práce začíná analýzou funkcí Polarionu následovanou recenzí nejběžnějších ML frameworků a algoritmů vhodných pro použití v prostředí Polarionu. Následuje popis důležitých případů použití s cílem najít nejvhodnějšího případu užití pro aplikaci ML. Pro dva vybrané případy použití je vytvořen prototyp s využitím produkčních dat a následně je diskutována náročnost budoucího nasazení do produkce spolu s komplikací ve formě nové GDPR směrnice.

Klíčová slova Strojové učení, životní cyklus softwarových aplikací, ALM, GDPR, AWS, Amazon web services, LDA, Latent Dirichlet allocation

Abstract

Master thesis contains an overview of possible ways how to use Machine learning (ML) in Polarion ALM software. Thesis begins with analysis of Polarion functionality and is followed by a review of ML frameworks and algorithms suitable for using in Polarion environment. Followed by description of important user cases with business value appropriate for ML implementation. For two selected user cases is created a prototype based on production data.

Finally we are discussing deployment to production and its problems in the light of new GDPR regulation.

Keywords Machine learning, application lifecycle management, ALM, GDPR, AWS, Amazon web services, LDA, Latent Dirichlet allocation

Contents

Introduction	1
1 The aim of this thesis	3
2 Polarion	5
2.1 ALM	5
2.2 A unified solution	6
2.3 Development process in complex or regulated environments . .	10
2.4 Accelerate collaboration	12
2.5 Medical domain	14
3 Machine learning	17
3.1 TensorFlow	19
3.2 PyTorch	20
3.3 Keras	20
3.4 Caffe2	20
3.5 Amazon web services	20
3.6 Microsoft Cognitive Toolkit	23
3.7 Apache Spark MLlib	24
3.8 MxNet	25
3.9 Latent Dirichlet allocation	25
4 Polarion’s use cases for ML	27
4.1 OCR for image attachments	27
4.2 Advanced Search in Polarion	28
4.3 Suggestion for optimization of Polarion configuration	28
4.4 Suggesting how to split a document Space	29
4.5 Translating documents	31
4.6 Estimations and expectations	31
4.7 Defect recommendations	32

4.8	Chat bot	33
4.9	The most complained parts of Polarion	35
4.10	Detection of duplicated defects	35
5	Proof of concept prototypes	37
5.1	Prototype - Suggest how to split a Document Space	37
5.2	Prototype - defect recommendations	43
6	ML in Polarion production	47
6.1	ML future	51
6.2	Changes to environment	51
	Conclusion	55
	Bibliography	57
A	List of abbreviations used	59
B	CD contains	61

List of Figures

2.1	A unified solution for ALM [1]	7
2.2	Document workflow [1]	9
2.3	Companies expectation from using Agile[1]	11
2.4	Collaboration traceability workflow [1]	12
2.5	Medical risk management workflow [1]	15
3.1	Deep learning concept of Tensorflow [2]	19
3.2	AWS ML stack [3]	22
3.3	Amazaon machine learning stack [3]	23
3.4	CNTK architecture [4]	23
3.5	Spark ecosystem [5]	25
3.6	Plate notation (Bayesian inference) for LDA [6]	26
4.1	Attachments in Polarion	27
4.2	Search Work Items in Polarion	28
4.3	Document space in Polarion	30
4.4	Estimations in Polarion	32
4.5	Properties of defect work item in Polarion	33
4.6	Help page in Polarion	34
4.7	Similar defect suggestion in Polarion	35
5.1	Tree chart of documents	39
5.2	Topics distribution in external database documents	40
5.3	Topics distribution in variant management documents	40
5.4	Circle representation of all topics	41
5.5	Detail of topics in circle representation	42
5.6	Communication scheme	44
5.7	Practical use	45
6.1	GDPR consumer rights [8]	48
6.2	GDPR consumer rights in detail [8]	50

6.3	Expected IT budget for ML[7]	52
6.4	Technology trends over the next 3 years[7]	53

Introduction

“The revolution is just beginning, but it’s real – and the time to act is now. In fact, it is yours for the taking to harness a broad platform, services and ecosystem to transform your business. A unified approach to application lifecycle management is not a futuristic technology trend. It’s here today, and the good news is that you don’t have to completely stop and reset, but can smoothly transition from squeezing the most out of your existing business processes to making your organization thrive.”

Kurt Bittner
Analyst
Forrester Research

The author of this thesis has been working in software development for more than 10 years. During this time, he as explored the question of how to manage and keep up-to-date info, for a large project where more than a dozen people are involved. This is also why he chooses to work as a developer for Polarion.

We live in the world that is changing rapidly. No part of life can escape these changes, the software development sector included. It fact it needs to adapt on a daily basis to new requirements and technologies. This demand leads to a new level of management for software development, where tools, processes, implementation, testing and reporting are organized in a single place with continuously improving traceability and productivity. This comes hand in hand with automation in the form of ML that elevates the user experience and reporting to the next level. One such product is Polarion[1] and this work will analyze it’s use cases and find out what areas are good candidates for using ML techniques to improve the business value of the product.

The aim of this thesis

The aim of this thesis is to analyze and identify machine learning (ML) use cases that would prove valuable for Polarion's application lifecycle management (ALM) software. A proof of concept prototype will be supplied for each selected use case.

1. Analyze and describe Polarion in order to identify suitable use cases to apply ML to.
2. Provide a review of ML frameworks and algorithms that are relevant for such an application.
3. Describe several use cases for ML and define their benefit to both ALM as a business and the users that deploy it.
4. Choose a scenario from the previous investigation and implement a proof of concept prototype.
5. Discuss the possibility of the full implementation and deployment of the previous prototype into the production environment.

Polarion

Organizations are often struggling with the old processes of doing things. They focus on isolated process optimization instead of driving business value through comprehensive synchronization. With Polarion, customers have been able to get their teams out of their silos and orchestrate development efforts across the entire application lifecycle. This approach has empowered stakeholders to better perform tasks in context and quickly make sound decisions based on real-time access to information.

You can try Polarion on <https://polarion.plm.automation.siemens.com/>.

Now let's take a look what Polarion comes with and how it improves upon old fashion processes.

2.1 ALM

“ALM is a paradox in the software engineering world, where engineers recognize the need for requirements management, change and configuration management, QA and test management, and so on, but are not familiar with the term ALM. This is a serious problem because ALM is necessary to manage software complexity, and the rise of embedded software in engineered products needs mature management processes and tools.”

Michael Azoff
Principal Analyst

Polarion is an ALM enterprise solution that deals with modern-day challenges. It has emerged with the intent to fast-track innovation, while safeguarding quality, functional safety, and compliance. Its ability to speed up the development and delivery of innovative applications is becoming essential to the success of businesses in any industry.

ALM points to these three aspects:

1. An application has to be delivered as fast as possible and time itself is a new strategy weapon.
2. Information technologies are the fuel accelerating business success.
3. Errors are not forgiven and can go viral in an instant. QA rules must be set and obeyed.

2.2 A unified solution

Polarion is a single solution that provides all the tools needed to build whole application from the ground up. At the same time it ensures that data and logic are in a persistent state during the entire process.

This helps with regulations. This basic word means a lot in the world of software development, where processes are regulated by internal or external subjects and have an essential impact on the cost of a product, not only during its development, but after its release.

What ALM comes with?

The main advantages attributed to the ALM process (graphically shown in the picture 2.1):

- Agility through improved collaboration.
- Productivity through process integration.
- Auditability through traceability and accountability.
- Quality through transparency and automation.
- Innovation through unlocked team synergy.
- Predictability through better estimation and reporting.

Let's a look a little closer at some of these advantages.

2.2.1 Agility through improved collaboration

If faster time-to-market is a key success factor in today's competitive environment, real-time collaboration and the contextual performance of tasks are the means to stay ahead. In many cases, lightweight Agile software development methods have replaced or augmented incremental waterfall methods to release products more frequently.

Polarion provides flexible support for Agile or Lean, as well as traditional and hybrid environments, including any customized Scrum, feature-driven development, Kanban, extreme programming, or rational unified process methodologies.



Figure 2.1: A unified solution for ALM [1]

Polarion's 100 percent browser-based architecture makes information universally accessible from anywhere for any collaborator. Collaboration is easy and teams divided around the world can communicate to each other and solve tasks together.

2.2.2 Productivity through process integration

A large number of Polarion customers apply a combination of Agile and DevOps methodologies. Polarion's ALM solution is the perfect conduit for DevOps, allowing for the easy synchronization of development and delivery processes spanning requirements definition, feature development, quality testing, and maintenance. Any problem can be easily tracked back to its source and maintenance time is significantly reduced. (Right up to real-time fixes).

Polarion integrates with other tools. This is either done using an extension or through native integrations. This means that customers can still use their own tools and data repositories and then integrate them with Polarion. Polarion has been on the market for many years and in this time many extensions have been created by customers or professional services and placed on the official Polarion marketplace where they can be freely downloaded. The average customer will find all they with regards to third-party integration on Polarion's marketplace.

2.2.3 Auditability through traceability and accountability

“We chose Polarion ALM at Phoenix Contact in the Business Unit Automation to consolidate our very heterogeneous tool landscape – PVCS, Bugzilla, OneTree. With Polarion ALM we achieved transparency on all levels of development and we got fast acceptance in the teams. We now see exactly and in detail the status and the progress in our projects in the different project phases.”

Andreas Deuter

Phoenix Contact Electronics

Every change is stored. Polarion stores all you need to track down what, how and by whom a change was made. In an enterprise environment where hundreds of people are working, it is hard to keep in touch with who does what. A place where you see it all is priceless and helps minimize the risk of black holes where a task is left or forgotten without any notice. Such a missing task can have a very serious impact on cost or even the release date itself.

2.2.4 Quality through transparency and automation

A big problem with this approach is that team members usually get the information about what they need to accomplish from static documents that tend to go out-of-date as quickly as they were created. But perhaps worst of all, changes and ad hoc decisions often fail to take into account the downstream impact.

Polarion's processes provide way to track all requirement changes to help to keep in touch with the actual state of what needs to be done. For instance if a Product Owner changes a task, developers are informed and can react by accepting the task or request for more information about the change. In both cases, all sides know what is happening and what comes next.

The time when all operations were done by humans is over and automation plays important role in IT management. Polarion provides an environment to

run automatic jobs to build, test, check, ... or whatever a customer may need. Customers are free to implement their own job and run them in the same way as native ones.

2.2.5 Automating proof of compliance

One of the most important aspects is document workflow and it will be discussed in further detail, later in this thesis.

Think of a Polarion document like a Word document that contains a number of externally referenced tasks called 'Work Items'. Each document has its own text (headings descriptions etc.), but also contains the dynamic headings and descriptions of up to a million external, standalone Work Items. When another user updates the description of, for example, a requirement Work Item, then its description is automatically updated in any document that contains it. The reverse can also be done. With the appropriate permissions, someone can update a Work Item from within a document and it will update the standalone external Work Items and its description in all other documents that contain it. Let's look at the workflow shown in 2.2.

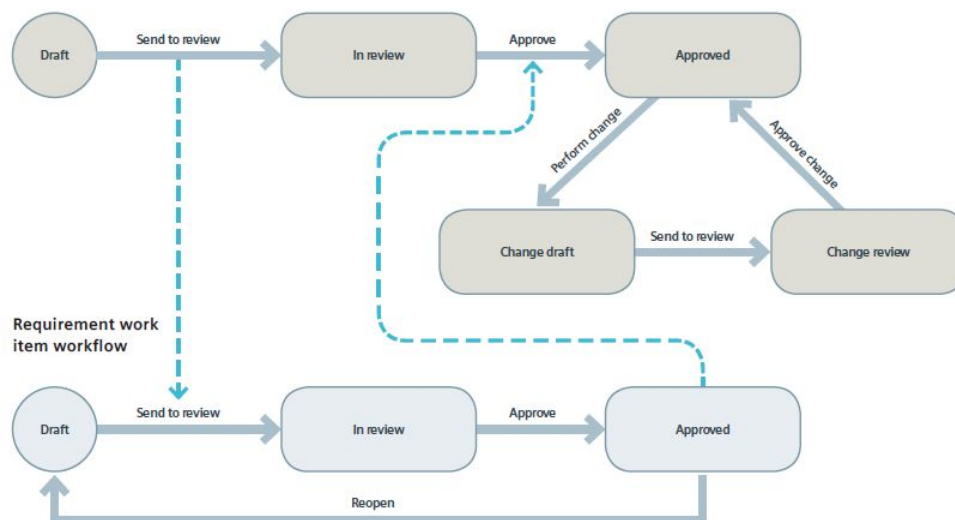


Figure 2.2: Document workflow [1]

Figure ?? shows two separate workflows. One for documents and one for Work Items. The workflows themselves do not need explanation but it is important to understand the relationship between them. A Document can not move to the next 'Status' state unless all its Work Items have also been moved to a desired state. This keeps traceability and collaboration consistent

by pushing a team to accept their work and minimizing undesired drafts, fakes and time wasted on unrelated work.

2.2.6 Innovation through unlocked team synergy

We live in an information age. Information is all around us and its hard to determine what is important for us or our company and what is not. A Team is a great way to share information but what if it has a lot of members? Polarion contains a way to cooperate and improve knowledge base as fast possible and share it with other co-workers. The time wasted solving an already solved issue is rapidly reduced and a new task can be done based on the results of previous work. This leads to more accurate estimations and risk assessment.

2.3 Development process in complex or regulated environments

Most collaborators don't have the unified tools environment necessary to get them on the same page at the same time, however, and the resulting disconnects have an increasingly negative impacts such as, Disruptive institutions with a record number of regulatory warning letters, product failures, recalls, legal sanctions, a loss in market position and the associated cost explosions.

Topping the list of challenges in the new world of software driven innovation is the need for a tight orchestration across disparate teams, growing regulatory demands and the increasing role of suppliers as innovation partners. Companies that are able to shift gears to meet the growing complexity will be well positioned to secure new market opportunities.

A common first step is to adopt the Agile approach. Companies have different expectations about its benefits and this is shown in the picture below 2.3.

The result is clear. Companies feel that IT is starting to be more and more complex and the demands from business is harder to satisfy. Polarion supports all Agile methods and is ready to help a client company improve its environment. More importantly its "Agile templates" can be modified based on a customer's demands to fit their needs and specific requirements.

"Siemens PLM Software's Polarion products presents the opportunity to allocate our complex and formal development rules via one state-of-the-art tool. The modularity and flexibility make the adjustment to our needs simple and effective. The traceability and workflow features are convincing and really assist the everyday activities."

Christian Kettl
MTU Aero Engines

2.3. Development process in complex or regulated environments

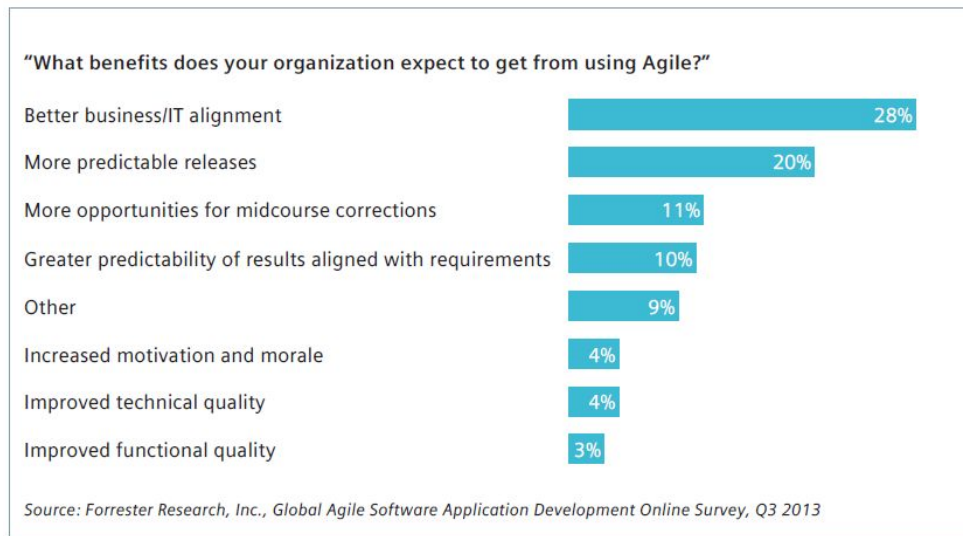


Figure 2.3: Companies expectation from using Agile[1]

2.3.1 Integration of ALM and PLM

Up until now we have been talking about ALM, but the main goal is to provide entire solution for product application lifecycle PLM where ALM is just a part of it. Where ALM is concentrated around applications PLM is the process of managing the lifecycle of a product itself. (From inception, creation to end of life.) Polarion is able to provide this functionality by integrating with external tools that can use Polarion as the source for the ALM work flow.

ALM-PLM integration benefits include:

- Integrated processes make cross-discipline synchronization very easy.
- Access to product and software requirements support comprehensive understanding of the product definition.
- Bi-directional linking enables cross-discipline lifecycle management and audit readiness.
- Change propagation and automatic notification enable comprehensive change impact analysis.
- Synchronized testing and reporting support cross-functional defect management.
- Linked, versioned data architecture without data duplication delivers closed-loop decision making.
- Integration makes holistic compliance reporting for every aspect of the manufacturing process a reality.

2.4 Accelerate collaboration

Development environments to synchronize team efforts have proliferated. But most of them are cobbled together, posing a wide range of disadvantages. Leveraging Polarion flexibility, customers can choose from different configurations to provide all collaborators with the level of information and functionality they need, while keeping the total cost of ownership the lowest in the industry.

These are the most common:

- Difficulty linking and tracing artifacts across differently structured repositories.
- Problems of low visibility into project status, impact of changes and release predictability.
- Lack of a cohesive feedback loop that brings important context to every stakeholder.



Figure 2.4: Collaboration traceability workflow [1]

2.4.1 The dilemma of requirements documentation

We need to be sure that all sides speak with a dialect that respects the specific domain. This requirement typically encompasses varying pieces of content, including:

- Paragraphs to provide overviews and explain details.
- Lists and tables to detail structured data and rules.
- Images and models to illustrate requirements.
- Flow charts to describe a series of events.

2.4.2 "easy-as-Word functionality"

If we say that the base object in Polarion is a Document we should expect that customers will use a document in the same way they use Microsoft Word. Fortunately the behaviour of Documents in Polarion is very similar in both use and appearance.

Customers can use known buttons and tools from Microsoft Word to edit and interact with a document and this speeds up the learning curve a lot.

2.4.3 Real-time access to content

Unlike of a Microsoft Word document, a Polarion document fully online and each change is immediately visible to everyone. Many users can simultaneously edit one item and Polarion then handles the merging of changes. This can of course lead to conflicts and when it does, users must handle the merge manually.

Consequently, companies that use Polarion Requirements are no longer forced to rely on meetings, sending emails, or circulating formal documents to make decisions, even with their partners and other external collaborators.

2.4.4 Tie in domain experts with their tools

As was said previously Polarion is expandable. Extensions can change or improve some existing functions, add completely new ones, transfer data from or to Polarion or connect it to external tools providing new functionality.

To complete the picture, connectors for popular third-party tools such as HP® Quality Center® and Atlassian® Jira® are available, and so is an open and fully documented Java API. As a result, a strong community of more than 100,000 members has formed and created extensions, integrations and customizations.

2.4.5 Deliver release predictability

Because every artifact change in the Polarion product is tracked and reported using the underlying configuration management system, customers automatically gain a complete audit trail of who did what, when and why, making it impossible to change anything without leaving a trace.

The “Visual Diff” functionality easily detects the changes between different states, and customers report that teams that take advantage of change management and impact analysis are much more successful.

2.4.6 Reduce time-to-market

All previous sections have one main purpose - to deliver a product in the shortest time but ensure its quality and traceability.

2.5 Medical domain

One of the strongest parts of Polarion is its ability to satisfy regulatory demands from various types of institutions that are making development more complex than before.

Lets talk about the medical domain.

Medical device product development work is a highly integrated and regulated process. Two key standards incorporated into medical device risk management are International Organization for Standardization (ISO) 14971:2009, which specifies the process for a manufacturer to identify the hazards associated with medical devices; and ISO Technical Information Report (TIR) 24971:2013, which provides guidance in addressing specific areas of ISO 14971 when implementing risk management. Europe has added to the mix with EN ISO 14971:2012, which is different in several important aspects, and is required if a company is selling medical devices in Europe. You can see this process in the Medical risk management workflow figure 2.5.

It is just an example how problematic and regulated this domain is.

To describe how exactly Polarion is used in the medical domain is beyond the scope of this thesis. But one point worth discussing is data. Data in the medical domain is a valuable asset whose use is restricted by law. Any legal violation has serious financial and social consequences. If we want to use this data we have to ensure that all rules are obeyed and the data is secure from theft or unauthorized use.

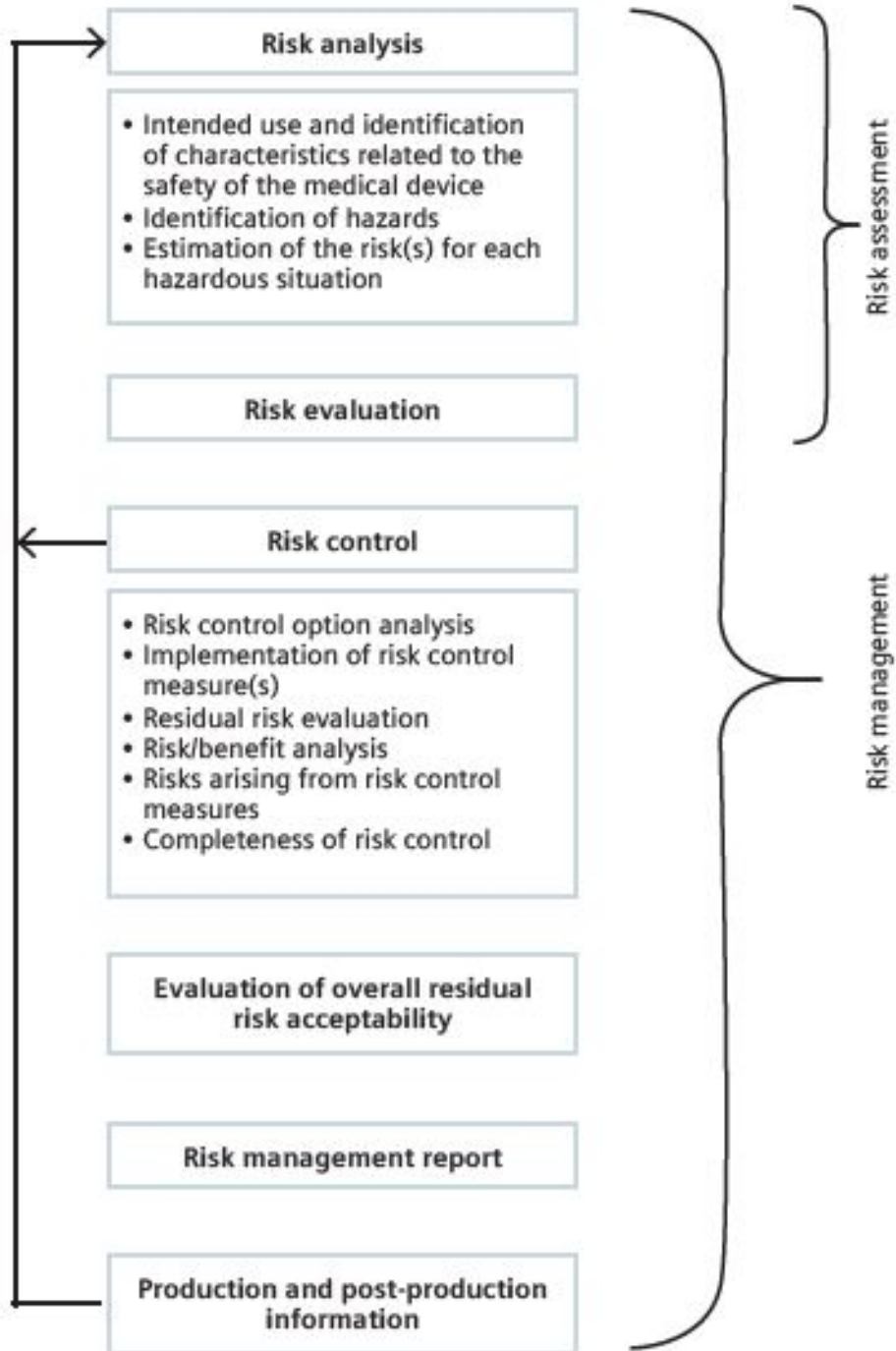


Figure 2.5: Medical risk management workflow [1]

Machine learning

It's been over 50 years since the first mention of ML by Arthur Lee Samuel in his study *Some Studies in Machine Learning Using the Game of Checkers*[9]. At that time it was just an idea and the IT industry was not ready to support it. But now we are fully in the information age. Machines are able to calculate incredibly fast and we are able to store all the data what we need. From this ML has arisen and companies from all domains have realised that ML can improve their product significantly, or even create entirely new ones.

The path from just a theory to practical usage was long and hard [10].

- 1979 — Students at Stanford University invent the “Stanford Cart” which can navigate obstacles in a room on its own.
 - If you look at it today, the achievement is a bit funny but, on the other hand, it is almost 40 years old.
- 1997 — IBM Deep Blue Beats Kasparov
 - A memorable event. It was also the moment when author of this thesis heard about ML for the first time. (Along with most of the public.) Machines stopped being perceived as simply calculators and earned the status of a thinking thing.
- 2016 — Beating Humans in Go
 - Google's AlphaGo was able to beat a professional human player using a combination of machine learning and tree search techniques. What will be next?

During MLP 2018 [11] presented a heartbreaking use case for facial recognition. Try to imagine a natural disaster that leads to thousands and thousands of refugees. Families are divided and with all the chaos, it is impossible to find each other. Depression is rising and may lead to panic or violence. ML offers a simple idea. Just take a picture of yourself and it will find your relatives. How? ML has learned from data from other families and has determined what common family characteristics are and how they can be used to quickly create family matches. Based on this, ML is able to say with high probability if someone is related to someone else.

What makes this example so special? This use case utilizes ML in a new way but the algorithms and frameworks were already here. Data was provided, a model was created and it works. Sounds pretty easy and of course in reality it was a bit more complicated, but it gives us a good idea of the current state of ML. Companies can use existing implementations without the need for a deep knowledge of complicated algorithms. They can start right away on their own problems and business cases and see results in a very short time.

It is said that when using existing solutions a PoF should not take more than 2 months according to learning task and processed data.

Learning can be done in different ways:

- **Supervised learning:** Relies on data where the data is annotated and an algorithm goes through it and tries to find similarities. For example, if we want to distinguish between pictures of cats and dogs, an algorithm will go through a lot of annotated pictures and classify them. After that, the algorithm will be able to correctly annotate new pictures based on what it's learned. The main disadvantage is the need for a large amount of preprocessed data.
- **Semi-supervised learning:** An algorithm does not have any annotations. Again a large amount of data is provided to the algorithm along with some characteristics of what we want to find. In the case of cats and dogs, the goal is to separate the pictures into two groups. During this separation the algorithm will learn rules about what cats and dogs look like. Compared to the previous example, only raw data is required and no preprocessing is needed.
- **Reinforcement learning:** The game of chess is a good example of this type of learning. When the goal is to win. An algorithm gets a set of scenarios with moves and results and it learns by trying to find the best moves that will lead to victory. With games, this approach has a great advantage because the algorithm can play far more games in a short period of time than a human player could.

For the first, let's see what ML frameworks or libraries are available.

3.1 TensorFlow

TensorFlow[12] is an open source software library with strong support for machine learning and deep learning. The flexible numerical computation core is also used across many other scientific domains. It was developed by Google for user internally but fast became one the most used ML frameworks. The reason for this is the ease with which developers can build and deploy applications.

Its main focused is on deep learning and it has more tools to support reinforcement learning and other algorithms. The Deep learning concept is shown in figure 3.1.

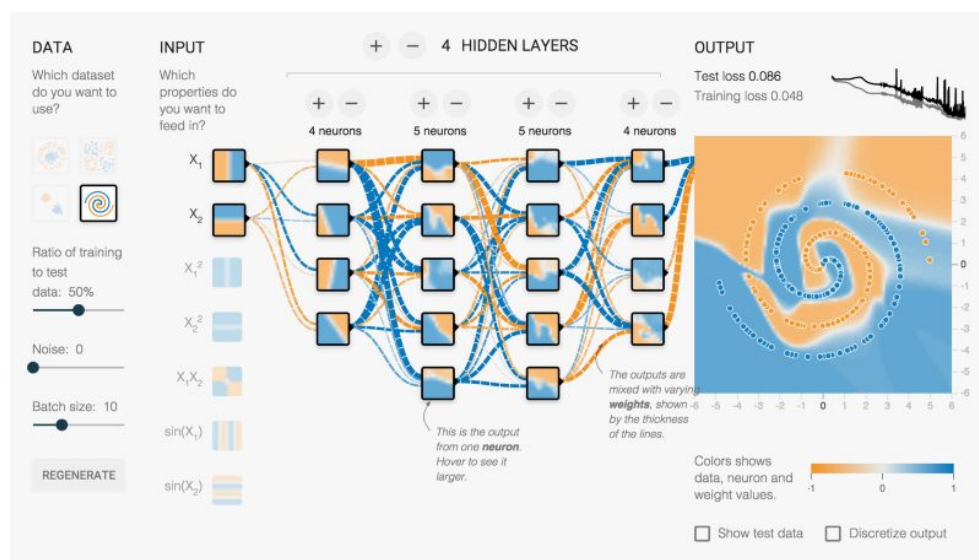


Figure 3.1: Deep learning concept of Tensorflow [2]

Tensorflow has much better performance and customizability than some other options, but its learning curve is much steeper, since you are not just plugging in data and labels into a constructor, but are actually creating the layers that will make up your model. To use it, you need a solid understanding of machine learning and mathematical concepts. (Especially linear algebra and calculus.)

3.2 PyTorch

A Python version of Torch, known as Pytorch, was open-sourced by Facebook in January 2017. It is based on dynamic computation graphs. This has a lot of useful benefits for certain types of RNN, situations when you need to generate weights and things where the very structure of the network changes.

PyTorch[13] also has a really nice interface and has the support of Facebook. It contains a lot of modular pieces that are easy to combine. The downsides are that it's a relatively newer framework, so there's not many integrations with it, not much of a community and not many papers implemented in it. (The dynamic computational graphs will also make many things rather inefficient because of the lack of static optimizations.)

3.3 Keras

Keras was created by Francois Chollet, a software engineer at Google and is a deep-learning library. It sits atop TensorFlow and Theano, providing an intuitive AP that is inspired by Torch and its development is fast growing.

While Keras[14] excels in rapid prototyping it lacks flexibility.

3.4 Caffe2

Caffe2[15] with a C++ engine is a successor to the original Caffe and is the second deep-learning framework to be backed by Facebook after Torch/PyTorch. The main difference is that Caffe2 is more scalable and light-weight but also rather limited in flexibility. (It is good for smartphone inference.)

3.5 Amazon web services

AWS provides a low-cost, scalable and highly reliable infrastructure platform in the cloud. This has been adopted by thousands of businesses globally. Australia, the US, Japan, Europe, Singapore and Brazil are among the data center locations. The locations are widespread to make sure the system is robust and secured against the impact of outages or other such problems.

Advantages[16]

- Security.
 - AWS conducts regular audits to ensure that its infrastructure is secure. It has implemented best practices in security and also provides documentation on how to deploy the security features. It ensures the availability,

integrity and confidentiality of data and provides ‘end to end’ privacy and ‘end to end’ security.

- Cost-Effectiveness
 - Users consume only as much storage or computing power as required. No upfront investment or minimum expenditure is required. Generally, it is not easy to predict the requirements for the resources, so a user might allocate fewer resources than required and impact customer satisfaction or allocate excessive resources and not be able to maximize a return on investment (ROI).
- Flexibility and Openness
 - Users can use the programming languages, architectures, operating systems and databases they are familiar with. In this manner, there won’t be any need for their IT personnel to pick up new skills and the overall time to market and productivity will improve.
- Elasticity and Agility
 - Allows users to experiment and innovate quickly through its huge global cloud infrastructure. They can quickly scale up or scale down on the basis of demand. They can also use new applications, rather than wait for months for hardware.

[17]: Common company issues avoided by using uAWS

- Overspending on hardware and storage capacity.
- Business leaders want IT to help preserve cash.
- A non-standardized IT environment and platform is expensive from a security, support and training perspective.

All these problems are solved by AWS.

AWS provides a wide range of services shown in the picture below 3.2.

3.5.1 Frameworks and infrastructure

At the low end, computation customers can choose from the most used frameworks:

- Mxnet
- TensorFlow
- Caffe2
- Keras
- CNTK
- PyTorch
- Gluon

Using them is just a matter of calling a united interface without a dependancy on the framework chosen. They are great for testing different approaches in a short time.

AWS ML Stack

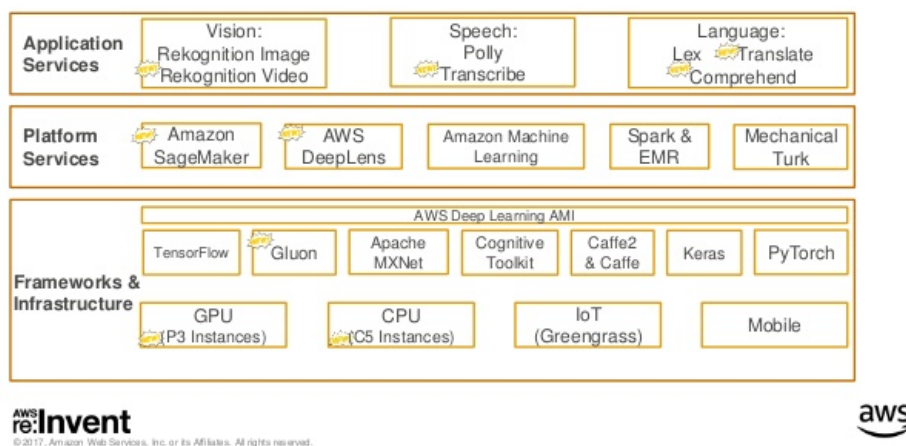


Figure 3.2: AWS ML stack [3]

3.5.2 Application services

It is an SaaS layer with applications ready for use. Customers do not need to implement their own solutions to test their data in common cases and can start to use it immediately.

3.5.3 Platform services

On this level resides what we can expect, except one, Amazon SageMaker.

Amazon realized that developers need a fast way to build, train and deploy machine learning models with as little effort as possible. So they created SageMaker.

Let's take a closer look at Amazon SageMaker.

3.5.4 SageMaker

Amazon SageMaker is a fully-managed platform that enables developers and data scientists to quickly and easily build, train, and deploy machine learning models at any scale with an easy to use GUI.

Amazon SageMaker runs on a fully managed elastic compute server. This relieves the data scientist or developer from DevOps concerns. Amazon SageMaker takes care of all the health checks, and outline infrastructure maintenance tasks via the built-in "Amazon CloudWatch monitoring and logging" service. Machine learning algorithms are provided pre-optimized, specifically enhanced to run on Amazon's compute servers. All customers have to do is

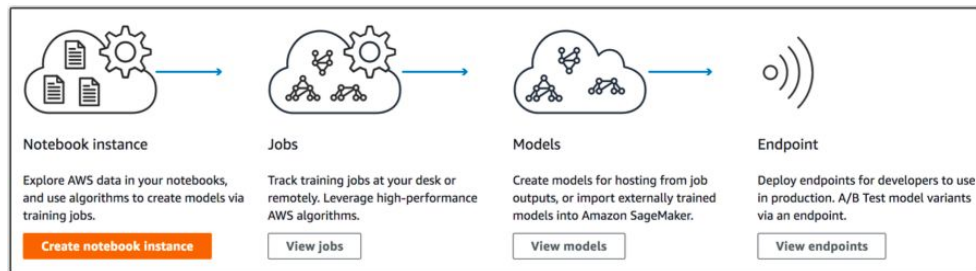


Figure 3.3: Amazon machine learning stack [3]

connect them to their data source. Trained models can be deployed for production directly from Amazon SageMaker. It deploys the model as well as implements a secure HTTP endpoint for the application. (Again from customers point of view, DevOps are not needed.) Last but not least, billing is based on utilization that is mostly dependent on individual customer use-cases and demands.

3.6 Microsoft Cognitive Toolkit

The Microsoft Cognitive Toolkit[4] written in C++ is a unified deep learning toolkit that describes neural networks as a series of computational steps via a directed graph. CNTK lets users easily realize and combine popular model types. CNTK has been available under an open-source license since April 2015 and is Microsoft's response to Google's TensorFlow.

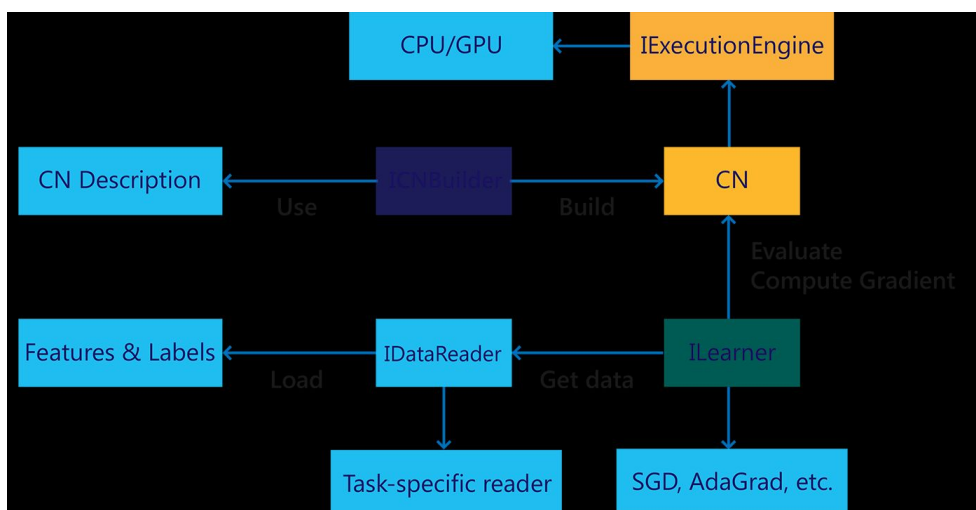


Figure 3.4: CNTK architecture [4]

Thanks to its architecture it is very flexible, allows for distributed training

and supports all the main programming languages. A downside is that it lacks visualizations.

3.7 Apache Spark MLlib

MLlib[18] is Apache Spark's scalable machine learning library.

Common[?] The problem with prototyping in Python or R is that moving from development to production environments requires extensive re-engineering.

For this Spark provides data engineers and data scientists with a powerful, unified engine that is both fast (100x faster than Hadoop for large-scale data processing) and easy to use. This allows users to solve their machine learning problems interactively and at a much greater scale.

The advantages of MLlib's design include:

- **Simplicity** - Simple APIs familiar to data scientists coming from tools like R and Python. Novices are able to run algorithms out of the box while experts can easily tune the system by adjusting important knobs and parameters.
- **Scalability** - The ability to run the same ML code on a laptop and a big cluster seamlessly without breaking down. This lets businesses use the same workflows as their user base and data sets grow.
- **Streamlined end-to-end** - MLlib is on top of Spark and it makes possible to tackle these distinct needs with a single tool instead of many disjointed ones. The advantages are lower learning curves, less complex development and production environments, and ultimately shorter times to deliver high-performing models.
- **Compatibility** - Data scientists often have workflows built up in common data science tools, such as R, Python and so on. MLlib provides tooling that makes it easier to integrate these existing workflows with Spark.

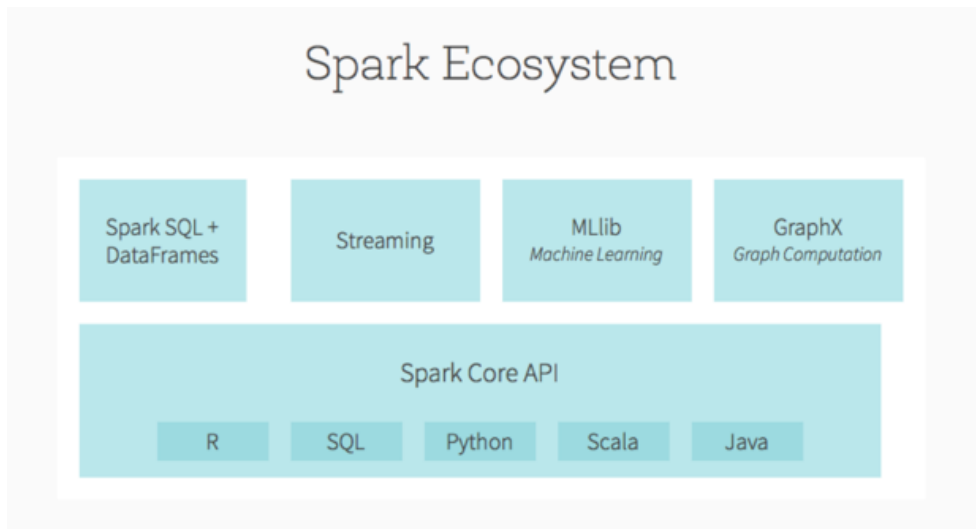


Figure 3.5: Spark ecosystem [5]

3.8 MxNet

[19] Like the previous frameworks Apache's MxNet is a modern open-source deep learning framework used to train, and deploy deep neural networks supporting multiple programming languages.

It supports an efficient deployment of a trained model to low-end devices for inference, such as mobile devices, IoT devices, Serverless or containers which should use models trained on higher-level environments because of their limited CPU and RAM resources.

It has been chosen by AWS to be part of their ML on demand infrastructure.

Now let's switch to a specific algorithm.

3.9 Latent Dirichlet allocation

LDA[20] is a type of topic modeling algorithm. The purpose of LDA is to learn the representation of a fixed number of topics, and given this number of topics learn the topic distribution that each document in a collection of documents has.

In LDA, each document may be viewed as a mixture of various topics. The sparse Dirichlet priors encode the intuition that documents cover only a small set of topics and that topics use only a small set of words frequently. In practice, this results in a better disambiguation of words and a more precise assignment of documents to topics.

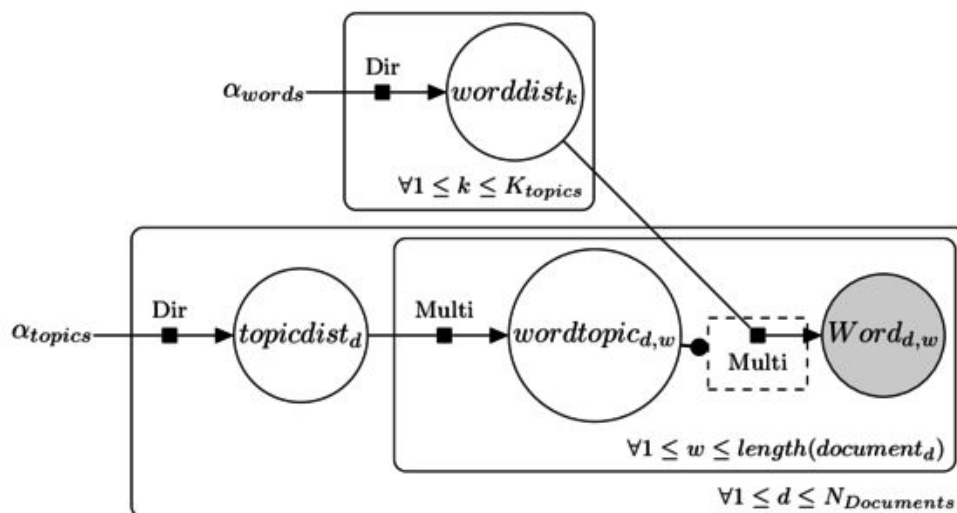


Figure 3.6: Plate notation (Bayesian inference) for LDA [6]

First we need to select the number of topics to discover. LDA will go through each of the words in each of the documents, and it will randomly assign a word to one of the K topics. Then it will have topic representations, how the words are distributed in each topic, and documents represented in terms of topics. This random form is not very optimal or accurate. To better this representation LDA will analyze per document the percentage of words within the document that were assigned to a particular topic. And for each word in the document, LDA will analyze the percentage of times that particular word has been assigned to a particular topic in all the documents.

Polarion's use cases for ML

Polarion enables its customers to plan and evaluate their every day work and if something is used frequently and periodically then there should be places for ML to improve it. Let's find some use cases that can be good candidates for ML.

4.1 OCR for image attachments

Polarion allows users to upload files on Work Items (see the picture below 4.1) Currently image files are treated without processing their contents and it has an impact on functions like search capabilities that can not use information within files. For this case OCR should serve well.



Title	File Name	Size	Author	Last Modified	Actions
screenshot-20180219-102	screenshot-20180219-102504.png [direct link]	182 B	Ondřej Chylik	2018-03-02 11:59	Show revisions
screenshot-20180219-102	screenshot-20180219-102726.png [direct link]	182 B	Ondřej Chylik	2018-03-02 11:59	Show revisions
screenshot-20180219-102	screenshot-20180219-102726.png [direct link]	258 B	Ondřej Chylik	2018-03-02 11:59	Show revisions
screenshot-20180219-102	screenshot-20180219-102726.png [direct link]	182 B	Ondřej Chylik	2018-03-02 11:59	Show revisions
screenshot-20180219-102	screenshot-20180219-102726.png [direct link]	264 B	Ondřej Chylik	2018-03-02 11:59	Show revisions
screenshot-20180219-102	screenshot-20180219-102726.png [direct link]	182 B	Ondřej Chylik	2018-03-02 11:59	Show revisions
screenshot-20180219-103	screenshot-20180219-103040.png [direct link]	258 B	Ondřej Chylik	2018-03-02 11:59	Show revisions

Figure 4.1: Attachments in Polarion

What is OCR? OCR is a technology that enables you to convert different types of documents, such as scanned paper documents, PDF files or images captured by a digital camera into editable and searchable data.

The advantages is that there are plenty of free OCR libraries in variety of programming languages that are ready to integrate with Polarion. The cost

of this implementation should not be high but it will lead to an increase in the index size that can rise rapidly due to the frequent use of attachments.

4.2 Advanced Search in Polarion

A common problem with a large amount of objects is being able to find what you are searching for. Polarion enterprise deployments can contain a large amount of such objects. For example customers have more than 10 000 different kind of Work Items and you have to know how to find a specific one. This should be improved to not only look for the exact same expressions but also for similarities and make suggestions based on the inputted data.

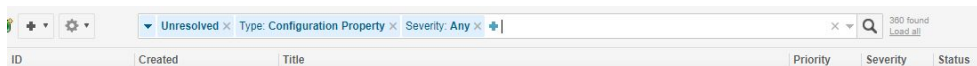


Figure 4.2: Search Work Items in Polarion

Work Item data can contain more specific challenges like references to other Work Items, pictures, graphs and stack traces. All these data types should be treated differently. Implementation will be difficult since the requirements are more unclear but results will lead to an improved user experience and less domain experts knowing the entire domain so all Work Items will be needed.

4.3 Suggestion for optimization of Polarion configuration

Each customer has special demands and a specific environment that Polarion runs on. Optimizing these configurations is not an easy task since only experts across multiple domains (DevOps, Polarion administrators, Polarion experts, ...) can see the impact and consequences of different settings.

This can be changed by us developers, since we know a wide range of cases where Polarion ALM (Polarion) or a customer was stuck and needed help. This experience involves looking into the log for a specific behaviour of use cases, memory/CPU/space/... consumption of Polarion or the environment itself. Based on this data we should be able to automatically recommend changes to a given configuration or at least start resolving problems more proactively before they become critical.

Example of log information:

```
2018-05-09 15:09:34,595 [main | u:p] INFO TXLOGGER - Summary for
'Platform startup': transactions: 0, DB: 20.1 s
[97% execute (1341x)] (1575x)
```

4.4. Suggesting how to split a document Space

```
2018-05-09 15:09:35,347 [main | u:p] INFO TXLOGGER - Summary for
'Context recognition': transactions: 0, svn: 0.436 s [49% getDir2
content (6x), 36% info (8x)] (15x)
2018-05-09 15:09:54,556 [LowLevelDataService-contextInitializer-3
| u:p] INFO TXLOGGER - Summary: Total: 19.2 s, CPU [user: 1.25 s,
system: 1.06 s], Allocated memory: 440.6 MB, transactions: 0, svn:
10.7 s [92% log2 (25x)] (428x), ObjectMaps: 1.59 s
[80% setLastProcessedRevision (451x)] (20180x)
2018-05-09 15:09:54,556 [LowLevelDataService-contextInitializer-5
| u:p] INFO TXLOGGER - Summary: Total: 19.2 s, CPU [user: 5.61 s,
system: 0.859 s], Allocated memory: 3.5 GB, transactions: 0, svn:
8.56 s [61% info (3043x), 28% getFile content (2015x)] (5093x),
ObjectMaps: 1.78 s [67% saveIfNeeded (24679x),
11% getPrimaryObjectLocation (6704x),
11% addLocation (13528x)] (121868x)
2018-05-09 15:09:54,556 [LowLevelDataService-contextInitializer-1
| u:p] INFO TXLOGGER - Summary: Total: 19.2 s, CPU [user: 0.891
s, system: 1.25 s], Allocated memory: 461.9 MB, transactions: 0,
svn: 12 s [94% log2 (45x)] (554x), ObjectMaps: 1.56 s
[77% setLastProcessedRevision (510x), 15% saveIfNeeded (5391x)]
(19414x)
2018-05-09 15:09:54,556 [LowLevelDataService-contextInitializer-4
| u:p] INFO TXLOGGER - Summary: Total: 19.2 s, CPU [user: 1.11 s,
system: 1.5 s], Allocated memory: 596.0 MB, transactions: 0, svn:
13 s [67% log2 (65x), 14% log (1x)] (1445x), ObjectMaps: 1.3 s [72%
setLastProcessedRevision (359x), 18% saveIfNeeded (6218x)] (22872x)
2018-05-09 15:09:54,556 [LowLevelDataService-contextInitializer-2
| u:p] INFO TXLOGGER - Summary: Total: 19.2 s, CPU [user: 1.55 s,
system: 1.8 s], Allocated memory: 705.2 MB, transactions: 0, svn:
13.6 s [88% log2 (56x)] (875x), ObjectMaps: 1.94 s
[79% setLastProcessedRevision (582x), 11% saveIfNeeded (8406x)]
(31707x)
2018-05-09 15:09:54,581 [main | u:p] INFO TXLOGGER - Summary for
'Context initialization': transactions: 0, svn: 57.9 s
[74% log2 (216x), 15% info (4897x)] (8395x), ObjectMaps: 8.17 s
[61% setLastProcessedRevision (1915x), 26% saveIfNeeded (49771x)]
(216041x)
```

4.4 Suggesting how to split a document Space

Polarion is about working with documents. It is expected that customers have a hundreds of documents separated into Document Spaces. Document Spaces

4. POLARION'S USE CASES FOR ML

are used as folders and make up a hierarchy. But in some cases customers create too many documents under one Document Space and the orientation within this space becomes unclear. The idea is to split documents under one document space into groups based on their topic. This is a job for LDA.

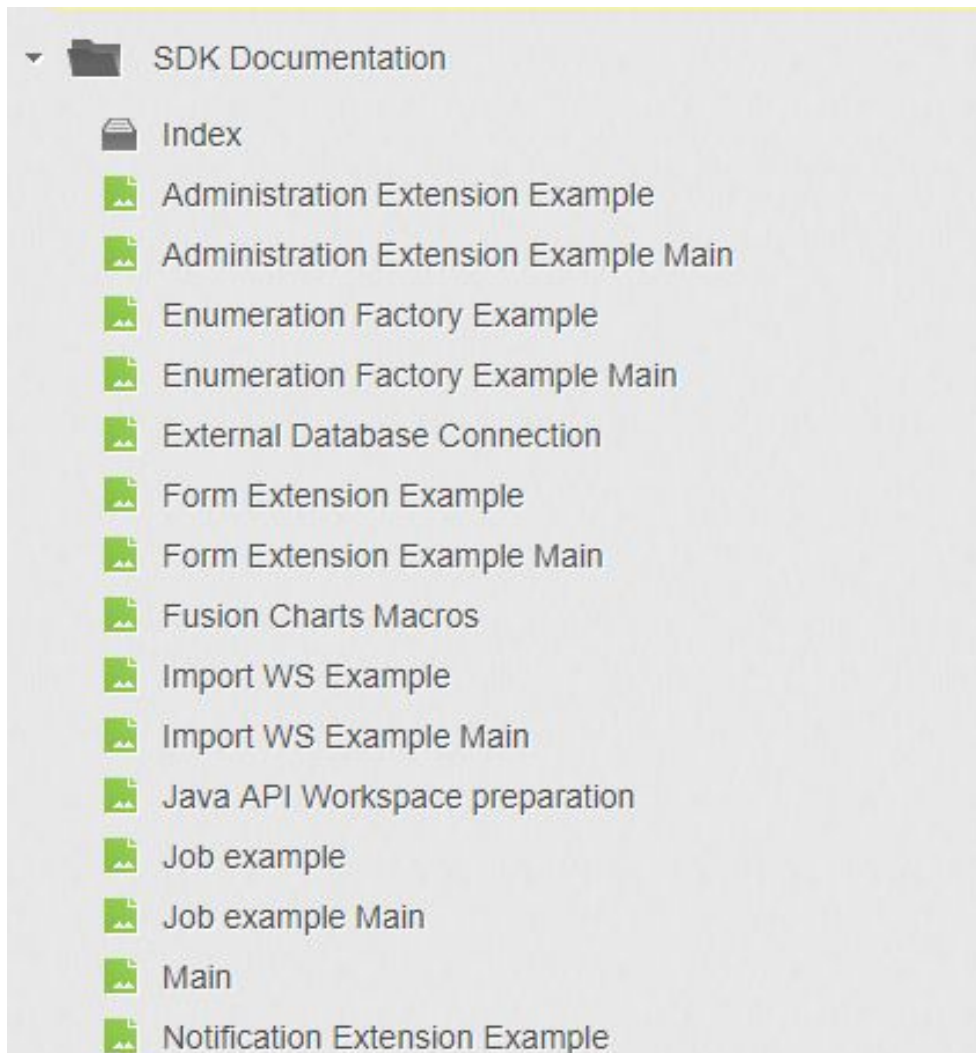


Figure 4.3: Document space in Polarion

As was said before, aside from plain text, a Polarion Document also contains Work Items that in turn contain their own text. (Including titles, descriptions, comments, and selected properties) All this information should be taken into account for topic creation. A Work Item can have a reference to another Work Item and so on, which raises the question, how many Work Items in such a chain is viable.

Adding topics into Polarion improves the user experience on a daily basis.

If we talk about splitting documents in one Document Space, topics can be also created for all documents and users will be able to find similar documents across Polarion, which is normally really hard task to do.

LDA is downloadable library and can be easily integrated into an existing code base. Changes on the side of Polarion will be much harder but the business value of this feature is good enough to balance it.

4.5 Translating documents

Normally customers have all their documents in one language. But there are cases when a document is created in one language and from it branched translations are required for external communication.

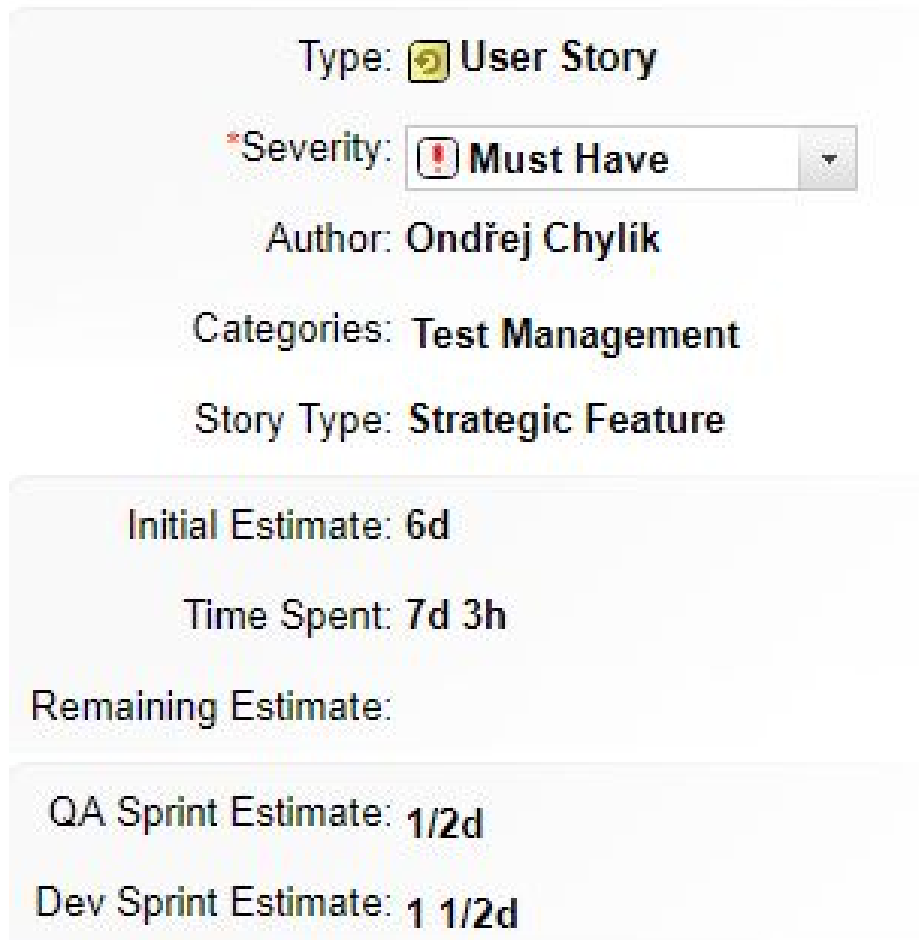
An important note is that a source document and a branched document are connected together, so that changes to one can be propagated to the other. The direction of propagation should be only from source to branched document because it is expected that a branched document will contain more information than its source. The first translation branched from a document has no problems but complications come when the source is changed and changes should be propagated to its branched documents. These documents can already have overwriting or context changing changes and the question is how to solve a collision issue.

A problem also lies in the translation itself. Although translation services are widely accessible, customer's data cannot be sent outside their environment, so a local translation server must be used.



4.6 Estimations and expectations

Agile planning is built around task estimation, fulfilling expectations and constant improvement by learning from estimations. The more experienced a developer is, the better estimates they make. The average time for a programmer job is in the order of years and the time to get familiar with Polarion is more than year. That is why estimates from inexperienced programmers are more guesswork than relevant data. On the other hand Polarion has at its disposal, the data from all previously completed projects and this data can be used to give hints on more accurate task estimations.

Firstly we need to find similar tasks. We can look for Work Items with the same parameter like category, story type and so on and combine it with the similarity of descriptions. All estimations from these found Work Items could be processed and shown to users to advise them on a more accurate estimate.



The image shows a screenshot of a work item in Polarion with the following fields and values:

- Type:  User Story
- *Severity:  Must Have
- Author: Ondřej Chylik
- Categories: Test Management
- Story Type: Strategic Feature

Initial Estimate: 6d

Time Spent: 7d 3h

Remaining Estimate:

QA Sprint Estimate: 1/2d

Dev Sprint Estimate: 1 1/2d

Figure 4.4: Estimations in Polarion

4.7 Defect recommendations

A Defect is a special type of Work Item in Polarion used to track errors and misbehaviours of the system. The creation of a Defect is done by whoever found the it, so the author needs to have knowledge about already existing defects and how to correctly set all parameters for that type of Defect. Polarion can try to find similar defects and help the user to better set properties up.

Similar defects will be searched based on the title of the defect that is written by the author and it must contain as accurate information as possible to ensure better results.

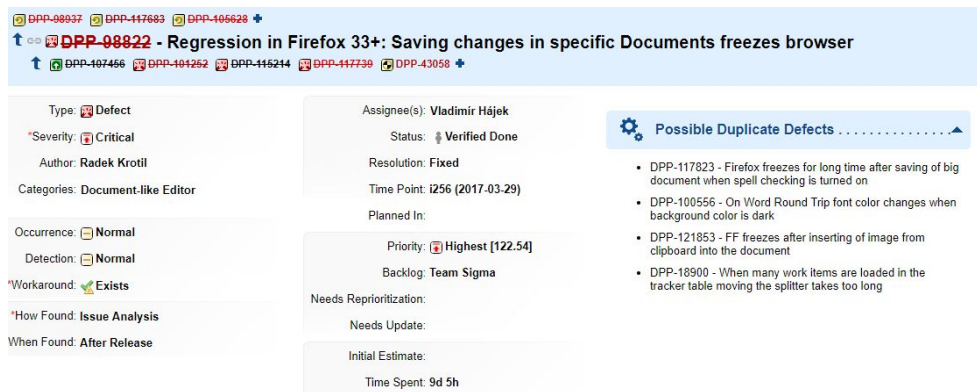


Figure 4.5: Properties of defect work item in Polarion

Properties for recommendations:

- severity
- priority
- assignee
- package
- category

Thanks to this, the responsible person will be informed of a new defect faster and the time from its discovery to the first attempt to fix it will be significantly shorter. It has clear business value and helps the entire company react faster and requires that less people be involved.

4.8 Chat bot

For such a huge product as Polarion it is natural that users are sometimes lost. (Either because they forget old stuff or spend time looking for new functions or processes.) In both cases they have two options. The first is to try search Polarion's Help pages ?? and the second is to bother a colleague. Because the help files are extensive, knowing how to frame the question determines how fast they can find what they need. Here's where a help chat bot could come in handy.

The idea is to take the existing help pages and index them in a usable format for some of the open-source chat bots. This first line of communication will help users solve their problems separately and questions to colleagues or even Polarion experts will be reduced. This will lead to a more productive environment.

4. POLARION'S USE CASES FOR ML

Search: GO [Search scope:](#) All topics

Contents

- 📄 **Polarion ALM Platform Help**
- 📄 Welcome
 - ⊕ 📄 Help and Other Documentation
 - 📄 Quick Overview
 - ⊕ 📄 About Polarion
- 📄 User Guide
 - ⊕ 📄 1. Portal Tour
 - ⊕ 📄 2. Projects
 - ⊕ 📄 3. Home Topic
 - ⊕ 📄 4. Dashboards Topic
 - ⊕ 📄 5. Plans Topic
 - ⊕ 📄 6. Work Items Topic
 - 📄 7. Documents and Pages Topic
 - ⊕ 📄 8. Test Runs Topic
 - ⊕ 📄 9. Baselines Topic
 - ⊕ 📄 10. Builds Topic
 - ⊕ 📄 11. Quality Topic & Metrics
 - ⊕ 📄 12. Reports Topic
 - ⊕ 📄 13. Monitor Topic
 - ⊕ 📄 14. Repository Browser Topic
 - ⊕ 📄 15. Working With Documents
 - ⊕ 📄 16. Working With Pages
 - ⊕ 📄 17. Polarion for Requirements Engineers
 - ⊕ 📄 18. Polarion for Project Managers
 - ⊕ 📄 19. Polarion for Developers
 - ⊕ 📄 20. Polarion for Testing and Quality Assurance
 - ⊕ 📄 21. Managing Variants
 - ⊕ 📄 Appendix
- 📄 Administrator's Guide
 - ⊕ 📄 1. Getting Started
 - ⊕ 📄 2. Administration Topics Overview
 - ⊕ 📄 3. Managing Users & Permissions
 - ⊕ 📄 4. Creating & Managing Projects
 - ⊕ 📄 5. Configuring Work Items
 - ⊕ 📄 6. Configuring Testing
 - ⊕ 📄 7. Configuring Plans
 - ⊕ 📄 8. Configuring the Portal
 - ⊕ 📄 9. Configuring Reports
 - ⊕ 📄 10. Configuring Documents and Pages
 - ⊕ 📄 11. Configuring Notifications
 - ⊕ 📄 12. Configuring SSL Support
 - ⊕ 📄 13. Configuring Building
 - ⊕ 📄 14. Advanced Build Management
 - ⊕ 📄 15. Configuring Repositories
 - ⊕ 📄 16. Configuring Connectors
 - ⊕ 📄 17. Configuration Properties
 - ⊕ 📄 18. System Maintenance
 - ⊕ 📄 19. Advanced Administration
- ⊕ 📄 **Polarion Reference**

34

Figure 4.6: Help page in Polarion

4.9 The most complained parts of Polarion

Polarion accepts complaints and suggestions every day from wide range of customers. As a results new projects are emerging to add new functionality or to fix the old ones to better satisfy customer requirements that were poorly analyzed during the first implementation. The creation of new projects are done by management welcome a tool that would track specific parts of Polarion that are targeted the most by customer requests and requirements.

We need to take into account not only information, but also its sentiment. (How much is a customer is really concerned with this problem.)

4.10 Detection of duplicated defects

Many new defects are created every day. The current process states that before the creation of a defect, a user should verify that the same defect does not already exist. This is done by searching Polarion and, as was said previously, it is not as good as should be, especially since there is no specific format on how to write defect descriptions and titles.

Polarion already contains basic duplicate analysis shown on 4.7.

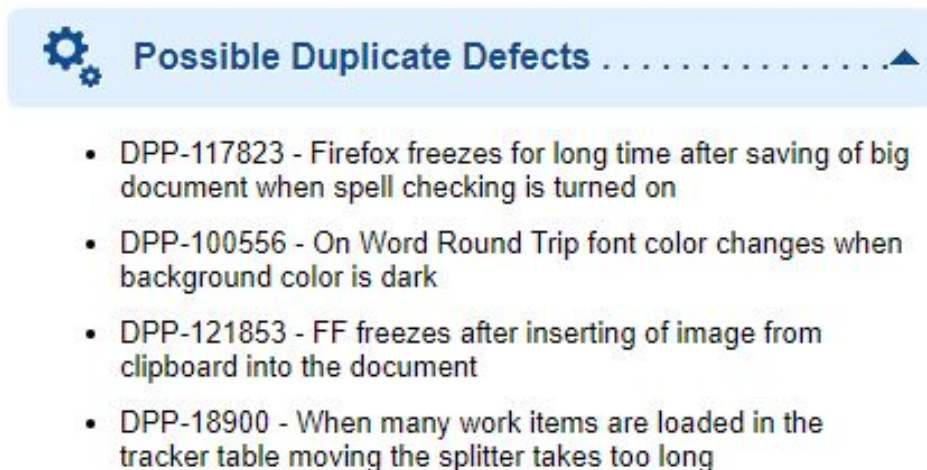


Figure 4.7: Similar defect suggestion in Polarion

An analysis of duplicates should be extended to more a complex level, also taking into account referenced Work Items, similar but not the same text, attachments and so on.

Proof of concept prototypes

We went through ML frameworks and algorithms and useful Polarion use cases expandable with ML. Now it is time to come up with PoF prototypes for some of the previous use cases.

From internal discussions we came up these two use cases:

- Suggest how to split a Document Space^{4.4}
- Defect recommendations^{4.7}

Let's move on to their implementation done in a team.

5.1 Prototype - Suggest how to split a Document Space

The task to split documents is a task to find their topics and group them accordingly. Finding topics is a perfect task for the previously mentioned LDA[?] algorithm.

Technologies used:

- PostgreSQL
- Java
 - Spring
 - LDA from <https://github.com/chen0040/java-lda>
[nosep]
- D3 Data Driven documents^[21]

[nosep]

The procedure is as follows:

1. data acquisition and clearing
2. creating topics
3. visualisation

For PoF purposes no integration with Polarion code base will be done.

5.1.1 data acquisition

It is possible to get data from Polarion through its API. But our case required to download data directly from database when data are stored to own database used just for the prototype. For this purpose the production database were cloned and used.

First step is to download all documents data. Then process each document's description and get list of work items contained in the document and download them. After it go through newly downloaded work items and look into their description for a references work items. If there is a referenced work items and download them if the work item was not downloaded previously.

Some problems were encountered:

1. Document has invalid data - reason for this are mainly historical data and these documents were skipped.
2. Incorrect referenced URL - link to referenced work item is in invalid format. Again reason for this is a historical form of link and these links were skipped.
3. Non existing work items - can happen and links were skipped

After application of these restriction we have thousands of documents with tens of thousands work items ready to next stage.

5.1.2 creating topics

This is where LDA library is coming to the scene.

Firstly we need to convert document data to just a plain text suitable for LDA. This processing takes text from document including its title and connects it with text from all his work items. Text from work items again means title plus description and if there is a referenced work item also text from it.

Processing is done in 4 levels when for the first level all documents are taken, topics generated. 5 most common topics are taken a documents suitable for them are put under them. Next level takes just a document under one topic a do the same thing as in previous step and so on for the next levels. Results is a three chart structure of all documents in Polarion??.

First runs revealed problem with interpretation of words so these sets were defined:

- stop words - common words used in language without specific meaning - ignored
- html - html tag commonly used in work item's description but inappropriate for topics modeling - ignored

5.1. Prototype - Suggest how to split a Document Space

- Polaron keywords - keywords from Polaron domain like *workitems*, *dpp*, *wiki* and so on - ignored
- special - command words like *while* or *for* - ignored

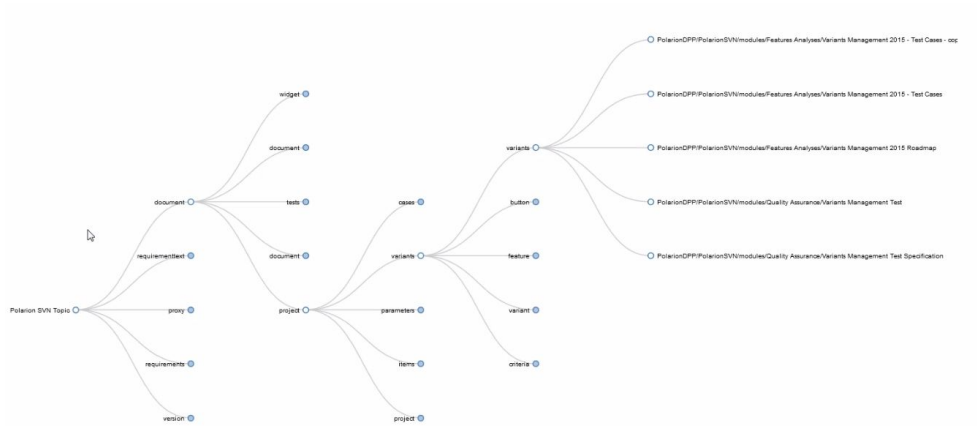


Figure 5.1: Tree chart of documents

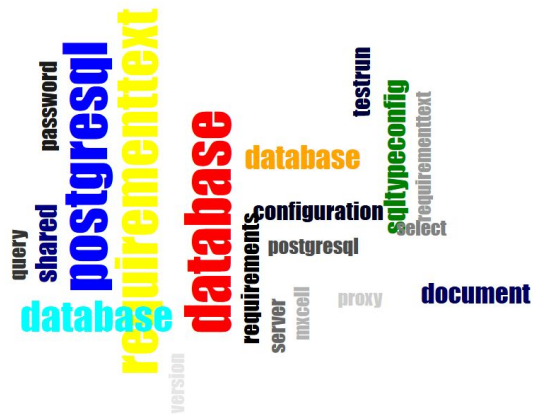
Using LDA library works flawlessly without any stuck or unexpected behaviour.

5.1.3 Visualisation

Using D3 library provides nice and easy to use visualizations.

Showing the most common topics for documents. More bigger the topic name is, the more topic is common in the documents. You can recognize that some names are redundant it is because topics were named after their most common word. See pictures below *Topics distribution in external database documents* 5.2 and *Topics distribution in variant management documents* 5.3.

Impressive visualization is a circle graph of all topics. It is circle view of previous tree chart where white circles are documents and depending on how big the white circle is, as a matter of fact, belongs to the topic. See *Circle representation of all topic* 5.4 and *Detail of topics in circle representation* 5.5.



Word cloud for topics in External database documentation.

Figure 5.2: Topics distribution in external database documents



Word cloud for topics in Variants management.

Figure 5.3: Topics distribution in variant management documents

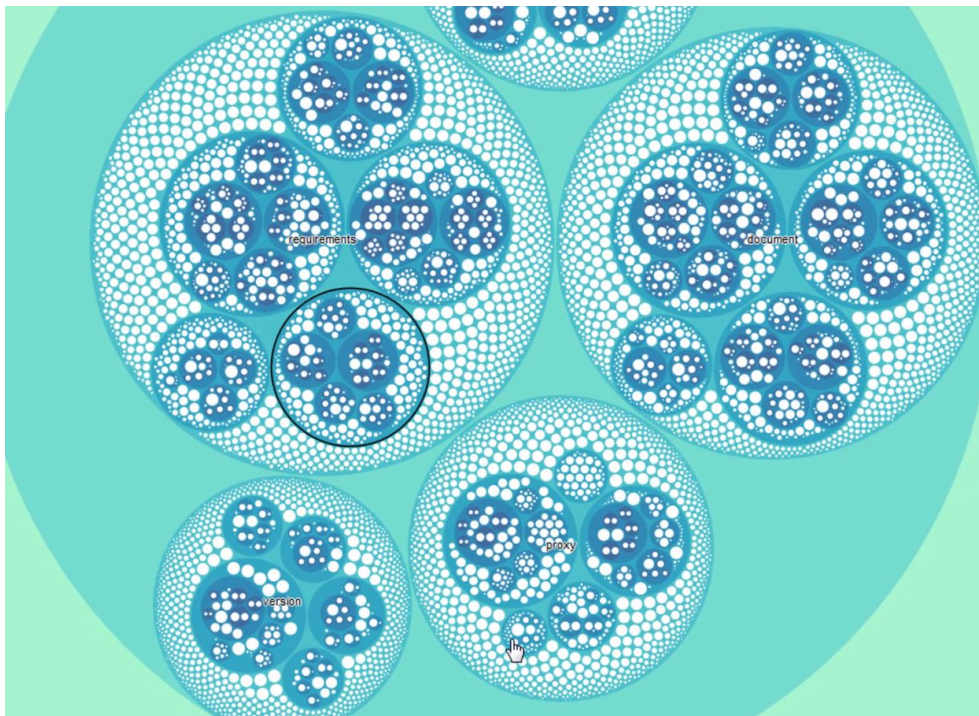


Figure 5.4: Circle representation of all topics

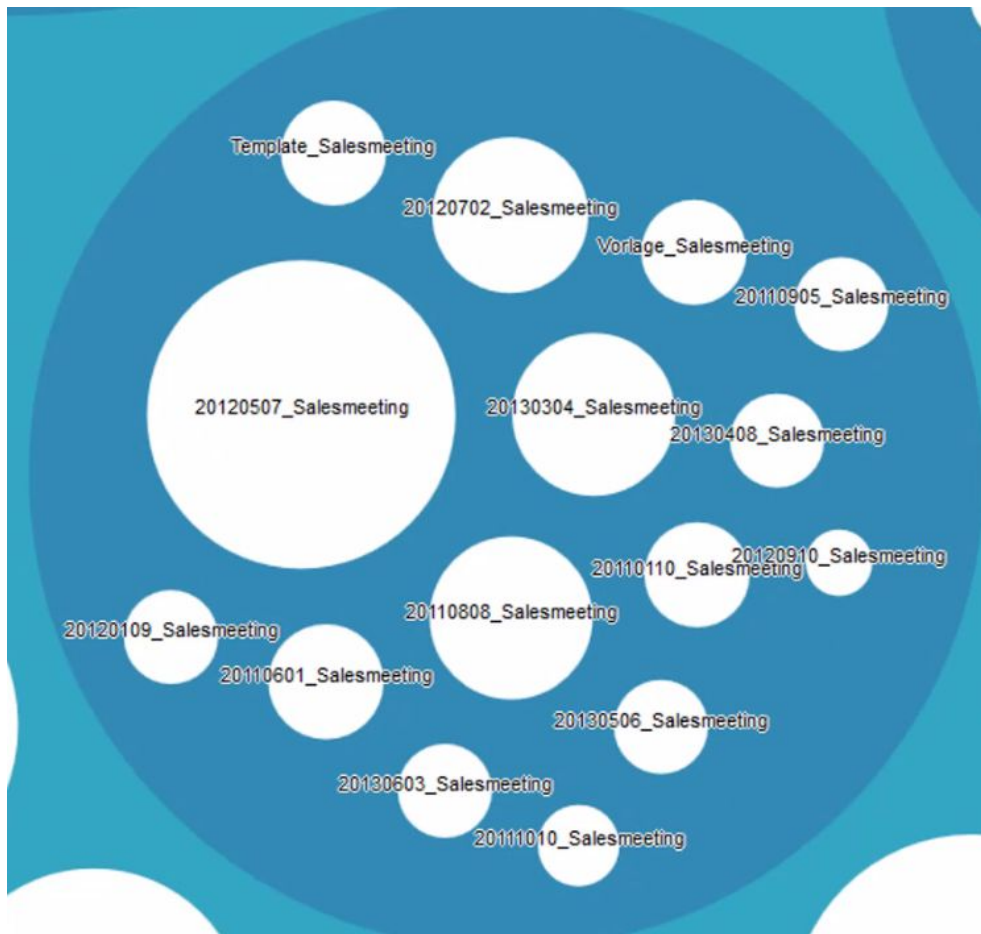


Figure 5.5: Detail of topics in circle representation

5.2 Prototype - defect recommendations

AWS was used for Machine learning (ML) implementation providing interface for data modeling in a form what is needed. Author also had a previous experiences and participate in the training around A (AWS) helping him to realize this prototype faster.

What was used:

- AWS cloud service and its own ML/AI engine.
- AWS provides a service to import data, store, train ML engine, and setup learning and ML service to cater to specific data sets.
- SCALA was used for a back-end and for extracting and pushing data from Polarion to AWS.
- JavaScript injection via Chrome extension was used to inject JS/HTML into a Polarion UI so you can see the results.
- Asynchronous real-time connection between Polarion UI, SCALA back-end and AWS was used as a connection.

5 categories were selected to recommendation:

- Assignee
- Priority
- Severity
- Category
- Package

Two separated parts were implemented:

- Loading Polarion defect data
- Real-time pull of results based on Polarion input

5.2.1 Loading Polarion defect data

Data was downloaded from Polarion services and processed to AWS cvs input format. Example of such a data file is below. In reality file has thousands and thousands of lines.

```
id,title,assignee,priority,severity,linkedRevisions
DPP-10188, linking a particular historical revision of a wiki
page does not work,Pavel Borovik,105.35,blocker,186728
DPP-10188, linking a particular historical revision of a wiki
page does not work,Pavel Borovik,105.35,blocker,191862
DPP-10110,Polarion program shortcuts do not works since Polarion
3.3 when updated from previous version,Jiri Banzsel,60.35,major,
188587
```

5. PROOF OF CONCEPT PROTOTYPES

DPP-10212,Export of formatted multi-lined text to Excel table fails with ClassCastException,Jiri Banzel,82.35,critical,186439

Then data file where pushed as a *AWS ML data set* into a AWS cloud for ML. ML uses 70 % of data to learn and then it test itself by using the rest (30 %) and provides a probability of its guesses by express it in %.

For each category was created one file and from this file one endpoint as so for each category estimation implementation consumes different endpoint.

5.2.2 Real-time pull of results based on Polarion input

Basic description is that user enters title of work item and click on ML bar to show recommendations. Communication scheme is shown in the picture *Communication scheme5.6*.

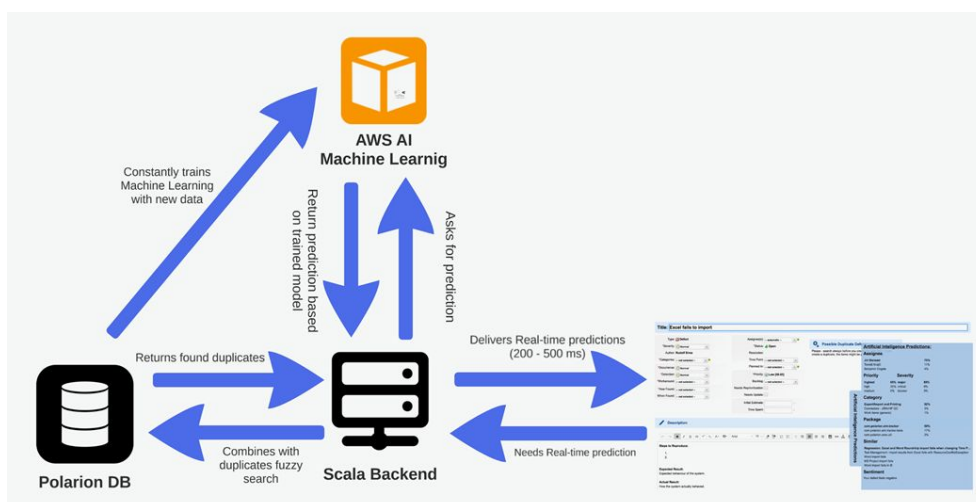


Figure 5.6: Communication scheme

1. User enters a title and request recommendations
2. Chrome extension grab title and send it to scala backend
3. Scala backend asks AWS for recommendations and returns it to chrome extension
4. Chrome extension shows recommendations to user as shown in the picture below 5.7

Quality of the results is surprisingly good and shows great potential for future production implementation. System is currently deployed on testing environment ready to be extended.

5.2. Prototype - defect recommendations

The screenshot shows a Jira issue form for a defect titled "Excel fails to import". The form includes fields for Type (Defect), Severity (Normal), Author (Radek Soukup), Categories, Occurrence (Normal), Detection (Normal), Workaround, How Found, and When Found. It also has fields for Assignee(s) (automatic), Status (Open), Resolution, Time Point, Planned In, Priority (Low [56.85]), Backlog, Needs Reprioritization, Needs Update, Initial Estimate, and Time Spent.

On the right side, there is a sidebar titled "Artificial Intelligence Predictions" which provides data for Assignee, Priority, Severity, Category, Package, Similar, and Sentiment.

Possible Duplicate Defects
Please - search always before you create a duplicate, the items might be

Artificial Intelligence Predictions:

Assignee	Percentage
Jiri Banzal	70%
Tomáš Krájč	11%
Benjamin Engle	4%

Priority	Severity	Percentage
Highest	major	85%
high	critical	9%
medium	blocker	3%

Category

Export/Import and Printing	92%
Connectors - JIRA/HP GC	3%
Work Items (general)	1%

Package

com.polarion.alm.tracker	59%
com.polarion.alm.tracker-tests	17%
com.polarion.core.util	3%

Similar

Regression: Excel and Word Round-4ip import fails when changing Time P...
Test Management: Import results from Excel fails with ResourceConflictException
Word import fails
MS Project import fails
Word Import fails in IE

Sentiment

Your defect feels negative

Description

Steps to Reproduce:

- 1.
- 2.

Expected Result:
Expected behaviour of the system.

Actual Result:
How the system actually behaved.

Figure 5.7: Practical use

ML in Polarion production

Both implemented prototypes are in test environment ready for further development together with not implemented ideas that are still on table. Decision to go to production is not so simple as can look like because if your product operate in a regulated environment any violation against these regulation may have fatal impact on Polarion trade mark. The main problem are data and customers must be sure that with data being handled correctly all the time.

Let's look at GDPR regulation coming into force just these days when this thesis is published.

6.0.1 General Data Protection Regulation

Data are fuel for ML but their use can be limited especially if we talk about customer's data with sensitive information. Now we face new regulation in form of GDPR that will change rules how to threat and store data. What GDPR is?

GDPR[22] is a legal framework that sets guidelines for the collection and processing of personal information of individuals within the European Union (EU). These GDPR sets out the principles for data management and the rights of the individual, while also imposing fines that can be revenue-based. The General Data Protection Regulation covers all companies that deal with data of EU citizens, so it is a critical regulation for corporate compliance officers at banks, insurers, and other financial companies. GDPR will come into effect across the EU on May 25, 2018.

Data types affected by GDPR are:

- Basic identity information such as name, address and ID numbers
- Web data (location, IP address, cookie data, ...)
- Health and genetic data

6. ML IN POLARION PRODUCTION

- Biometric data
- Racial or ethnic data
- Political opinions
- Sexual orientation

What companies will be affected by:

- A presence in an EU country.
- No presence in the EU, but it processes personal data of European residents.
- More than 250 employees.
- Fewer than 250 employees but its data-processing impacts the rights and freedoms of data subjects, is not occasional, or includes certain types of sensitive personal data. That effectively means almost all companies.

GDPR extends rights of data subjects that can be summarized like:

- the right to access
- the right to be forgotten, a.k.a. right to erasure
- the right to data portability

In more detail look at the picture below 6.1.



Figure 6.1: GDPR consumer rights [8]

- The data subject's right of access which means
 - the right to know whether data concerning him or her are being processed and
 - if so, access it with loads of additional stipulations.

-
- The data subject's right to rectification. When personal data are inaccurate, then controllers need to correct them indeed. The previously mentioned right to erasure or right to be forgotten with additional stipulations, among others if personal data has been made public.
 - The data subject right to restriction of processing. Simply said, the right of the consumer or whatever you call the natural person under the scope of the GDPR, to limit the processing of his/her personal data with, once more, several rules and exceptions of course.
 - The right to be informed. In general, the GDPR asks controllers and so on to inform data subjects on several matters. Providing clear and correct information is a key duty in many regards. Simply said, the GDPR wants consumers to know because if you don't know you can't decide. The controller must inform recipients who got these data, where feasible. And then the data subject also has a right, even if not strictly called a right, to ask *who are all these recipients who got to see my data*.
 - The right to data portability. This is again one of those data subject rights that are in the infographic and which we covered more in depth previously.
 - The data subject's right to object. That does indeed mean what it says: data subjects can say they don't want the personal data processing to be done or going on. Direct marketers and people who do profiling should pay a lot of attention to the right to object as it's a lot about them and certainly profiling with automated means.
 - The data subject right not to be subject to a decision based solely on automated processing, including profiling, which produces legal effects concerning him or her or similarly significantly affects him or her.

What we need to be careful about and is not part of current Data Protection Directive?

- Data breach notification: Controllers and processors are now required to notify supervisory authorities within 72 hours of learning of a breach and to notify the people to whom the data applies *without undue delay*.
- Explicit consent: GDPR requires that at the time you collect personal data, explicit consent must be given by the data subject. This means organisations can no longer bury generic consent in a long form full of legalese. Instead, organisations must offer specific information on what data is collected, how the data will be stored and processed, and must use clear and plain language. Nothing short of opt-in will do, and it must be as easy to withdraw consent as to give it.
- Data transfer out of the EU: Personal data must not leave the EU unless you have approval from the supervisory authority, or where the data subject is informed of the data transfer and associated risks and authorises the transfer.
- Data protection officer (DPO) appointment: If you process data on a

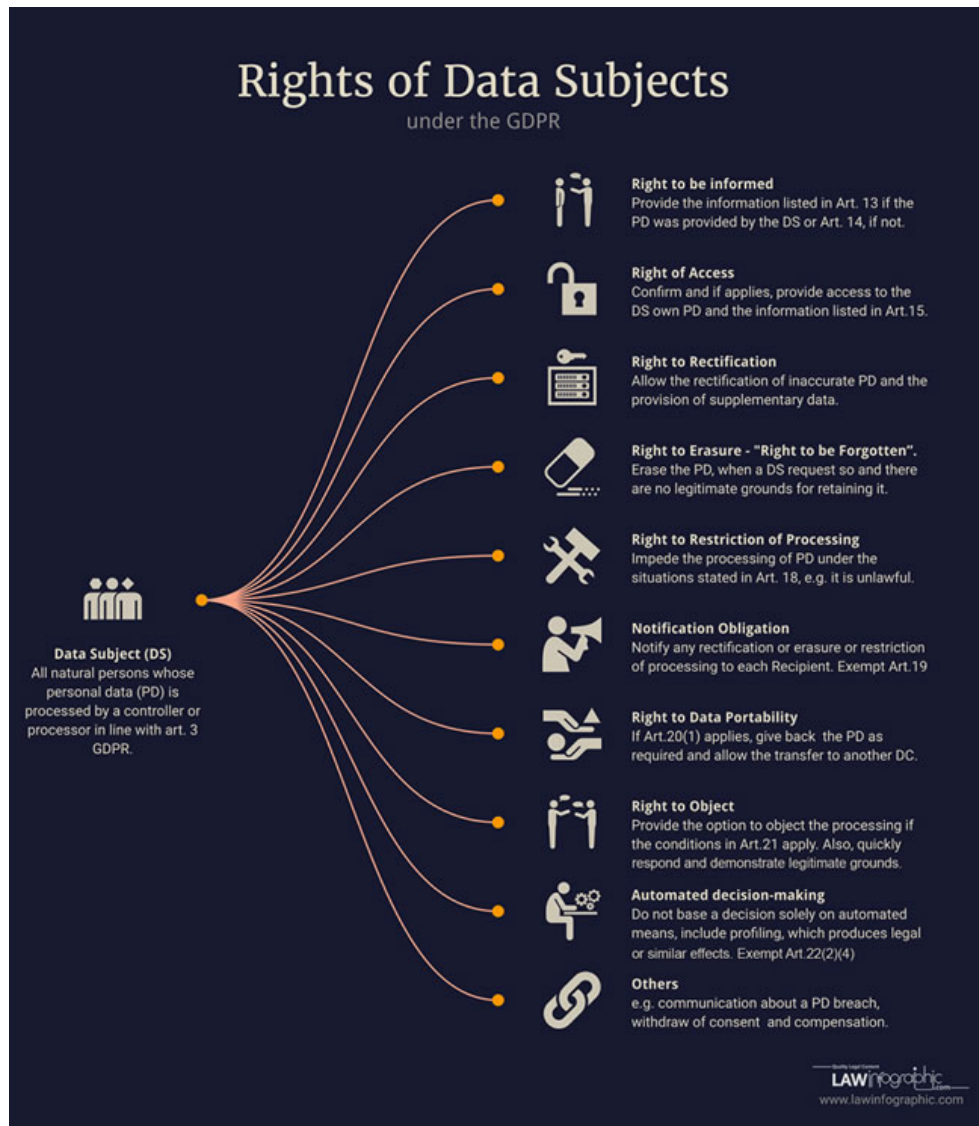


Figure 6.2: GDPR consumer rights in detail [8]

large scale then you must appoint, hire, assign or contract with a DPO, who is your representative to the supervisory authorities that monitor and ensure compliance with the regulation.

IT with ML is very competitive environment where companies around the world fight for primacy in research and business use. And as GDPR affects only EU this can disadvantage EU companies with their world competitors and their moving out of EU borders. Still true impact of GDPR is unclear and only future shows.

For Polarion ALM (Polarion) GDPR leads to more strictly regulation already so strict environment. Transfer data to external services seems to be quite hard and for this Polarion should implement its own ML solution that customers can install in their own secured environment and work with it without sending data to external repositories.

6.1 ML future

ML is no longer in category should have but it quickly move to technology that all system must have in some way and it does not matter if it will voice recognition, chat bot or some sort of data analysis. As you can see in the picture *Expected IT budget for ML*^{6.3} almost all companies invest in ML technologies with vision of improving their profit greatly.

Of course ML is not the only technology changing market and market expectations are seen in the picture *Technologies trends over next 3 years*^{6.4} and ML experts paid in gold.

An important change is the time it takes from an idea to PoF implementation. We are no longer talking about years but more likely months or even weeks or days. ML has evolved from pure mathematical algorithms to being more about the integration of existing ML solutions with your own data set. It helps management see and recognize the benefits of ML projects in their own product in a shorter time and with a lower cost. This combination is making it easier to start with production implementation.

6.2 Changes to environment

The practical impact of GDPR on data manipulation will become clearer as time goes on. It is better to wait and see what happens.

But to be clear, what we used for Defect recommendation^{5.2} can not be used in a production environment. We need to move the ML server to the Polarion environment by using the existing server's implementations, or create

6. ML IN POLARION PRODUCTION

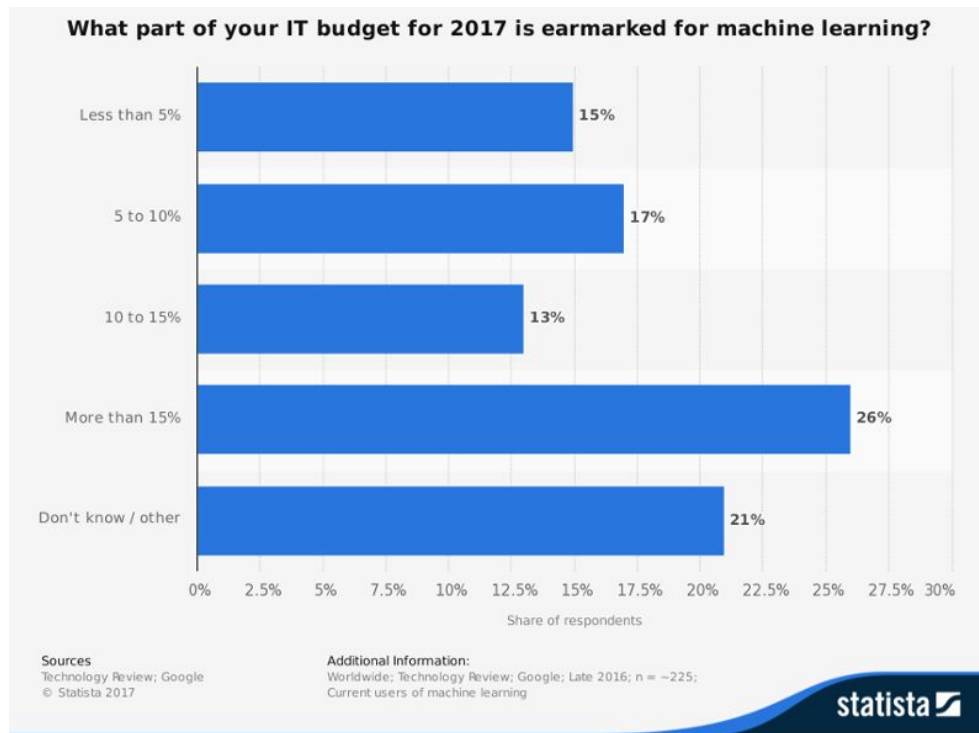


Figure 6.3: Expected IT budget for ML[7]

our own server. The ML community provides libraries and guides to help with the creation of our own server, and that is our next goal.

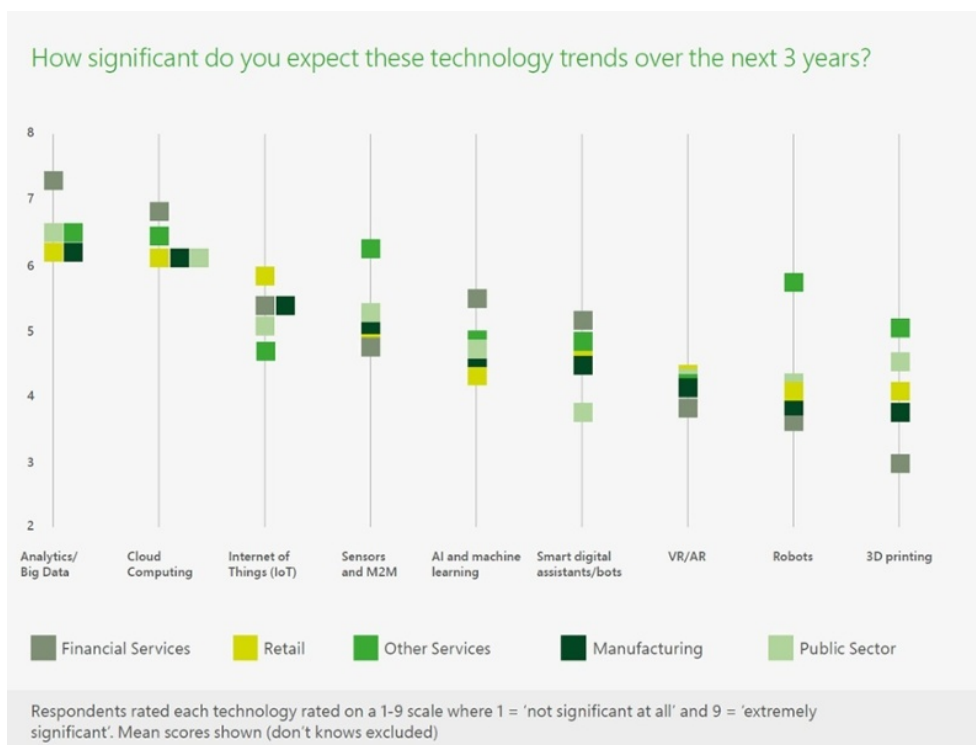


Figure 6.4: Technology trends over the next 3 years[7]

Conclusion

We have analyzed and described Polarion's core components and what makes it one of the best ALM solutions on the market for large enterprise clients with strong regulations to follow. That is why the Polarion user experience is worse in comparison with other, lightweight solutions.

A review of the existing ML solutions and algorithms revealed readiness of ML for fast prototyping and production deployment. Big companies like Facebook, Google or Microsoft continue with developing their own ML frameworks with growing community support and product deployments.

We dug deep into Polarion usage and found several promising user case candidates for ML prototyping. Each of these user cases has a valuable business potential moving Polarion one step above the competitors without the need for drastic changes to the current implementation.

We have chosen two use cases and successfully implemented a prototype for each of them. The implementations confirmed that prototyping is no longer task taking years, but instead months or even weeks with ML services like SaaS. Both prototypes are now in the test environment and ready for further development.

Despite the fact that prototyping is easier than ever before, a move to production remains complicated not only from the technology point of view, but also from a regulatory one. The new GDPR guidelines slows it down or even make it impossible.

Nevertheless ML will be a more and more common part of any application and companies have to catch up with this trend to survive. Polarion is working on it.

Bibliography

- [1] Polarion ALM. [online], [cit. 2018-05-01]. Dostupné z: <https://polarion.plm.automation.siemens.com/>
- [2] Deep Learning with Tensorflow Part 1 theory and setup. [online], [cit. 2018-05-06]. Dostupné z: <https://towardsdatascience.com/deep-learning-with-tensorflow-part-1-b19ce7803428>
- [3] Machine Learning on AWS. [online], [cit. 2018-05-06]. Dostupné z: <https://aws.amazon.com/machine-learning/>
- [4] The Microsoft Cognitive Toolkit. [online], [cit. 2018-05-06]. Dostupné z: <https://www.microsoft.com/en-us/cognitive-toolkit/>
- [5] Why you should use Spark for machine learning. [online], [cit. 2018-05-06]. Dostupné z: <https://www.infoworld.com/article/3031690/analytics/why-you-should-use-spark-for-machine-learning.html>
- [6] Experiments with Latent Dirichlet Allocation. [online], [cit. 2018-05-06]. Dostupné z: <https://mollermara.com/tag/statistics.html>
- [7] Feldman, M.: 10 Real-World Examples of Machine Learning and AI [2018]. [online], [cit. 2018-05-06]. Dostupné z: <https://www.redpixie.com/blog/examples-of-machine-learning>
- [8] Data subject rights and personal information: data subject rights under the GDPR. [online], [cit. 2018-05-06]. Dostupné z: <https://www.i-scoop.eu/gdpr/data-subject-rights-gdpr/>
- [9] Samuel, A. L.: Some Studies in Machine Learning Using the Game of Checkers. [online], [cit. 2018-05-02]. Dostupné z: <https://ieeexplore.ieee.org/document/5392560/>

BIBLIOGRAPHY

- [10] Marr, B.: A Short History of Machine Learning. [online], [cit. 2018-05-03]. Dostupné z: <https://www.forbes.com/sites/bernardmarr/2016/02/19/a-short-history-of-machine-learning-every-manager-should-read/>
- [11] Lanzetta, M.: Social Good at Cloud Scale.
- [12] TensorFlow. [online], [cit. 2018-05-05]. Dostupné z: <https://www.tensorflow.org/>
- [13] PyTorch. [online], [cit. 2018-05-06]. Dostupné z: <https://pytorch.org/>
- [14] Keras. [online], [cit. 2018-05-06]. Dostupné z: <https://keras.io/>
- [15] Caffe2. [online], [cit. 2018-05-06]. Dostupné z: <https://caffe2.ai/>
- [16] What is Amazon Cloud, Its Advantages and Why Should You Consider It. [online], [cit. 2018-05-06]. Dostupné z: <https://www.netsolutions.com/insights/what-is-amazon-cloud-its-advantages-and-why-should-you-consider-it/>
- [17] Benefits of Amazon Web Services (AWS). [online], [cit. 2018-05-06]. Dostupné z: <http://2ndwatch.com/blog/benefits-of-amazon-web-services-aws/>
- [18] Apache Spark MLlib. [online], [cit. 2018-05-06]. Dostupné z: <https://spark.apache.org/mllib/>
- [19]
- [20] Topic modeling with LDA introduction. [online], [cit. 2018-05-06]. Dostupné z: <https://algobeans.com/2015/06/21/laymans-explanation-of-topic-modeling-with-lda-2/>
- [21] D3 Data Driven Documents. [online], [cit. 2018-05-06]. Dostupné z: <https://d3js.org/>
- [22] General Data Protection Regulation (GDPR). [online], [cit. 2018-05-06]. Dostupné z: <https://www.investopedia.com/terms/g/general-data-protection-regulation-gdpr.asp>

List of abbreviations used

- ALM** Application lifecycle management
- Polarion** Polarion ALM
- ML** Machine learning
- pof** Proof of Concept
- RNN** Recurrent neural network
- AWS** Amazon web services
- GUI** Graphic user interface
- CNTK** Microsoft Cognitive Toolkit
- LDA** Latent Dirichlet allocation
- GDPR** General data protection regulation
- OCR** Optical character recognition
- D3** D3 Data-driven documents
- SaaS** Software as a service

CD contains

readme.txt	stručný popis obsahu CD
├ thesis	zdrojová forma práce ve formátu L ^A T _E X
└ text	text práce
├ thesis.pdf	text práce ve formátu PDF
└ thesis.ps	text práce ve formátu PS