Skill Sprints: disrupting IT consultancy and training

A holistic method for investing in employees.

He who binds to himself a joy
Does the winged life destroy
He who kisses the joy as it flies
Lives in eternity’s sunrise

Eternity - William Blake, (1757 - 1827)

Executive Summary

Every organisation faces challenges when given the options for technology adoption and digital transformation. These difficulties stem from ineffective communication amongst teams and regressive existing strategies that result in the shortfall of the final outcome. These obstacles prevent otherwise innovative or capable companies from reaching their true potential.

This white paper captures the most pressing issues that companies face when adopting new technology and implements solutions for a successful digital transformation.

Skill Sprint = technical consulting + instructional design

Skill Sprints are designed as a holistic approach to adopting new technology and transforming an organisation. Training usually is only concerned with individuals - with no attempt to transform the organisation. Consultants are typically disinterested in training. **Technology adoption and Digital Transformation require a process that takes care of the individual and organisational journey. Skil Sprints are designed to fill this gap / fill both these requirements.**

Investing in your employees is a very sensible thing to do!!!!

This white paper presents an argument against the current methods of technology adoption by exploring the concept of *tacit knowledge*.

Unlike explicit knowledge, which can be attained through instructions of information, tacit knowledge is gained through experience.[1]

In supporting the alternative of the current methods, the white paper will analyse the following topics:

- The crucial role of tacit knowledge for technical expertise

[1] Acquiring and Sharing Tacit Knowledge in Software Development Teams: An Empirical Study, Ryan & O'Connor, 2013

- The shortcomings of traditional training in building tacit knowledge
- The importance of Transactive Memory Systems (TMS)
- The effectiveness of pair programming and similar methods

The analysis is aimed at understanding the nature of “learning” itself and which methods are more effective to gather and utilise knowledge in an optimal way. In doing so, it will introduce a new solution for technology adoption: the *Skill Sprint*.

Current technology adoption and digital transformation strategies and their shortfalls

Companies currently choose between following strategies for technology adoption and digital transformation:

- Do it yourself
- Buy training
- Hire new staff
- Use consultants

Doing it yourself puts pressure on current staff to be able to deliver the unknown. This can be supplemented with training which is rarely adapted for the specific circumstance.

Alternatively companies will hire new staff or resort to consultants to move forward which results in a higher headcount or a costly dependency on a third party organisation. These methods all lack proper structure and defined deliverables which makes it difficult to even understand what a successful outcome could look like.

These methods all neglect the real potential of existing employees and lead to an uncertain return on investment.

Skill Sprint - the optimal solution for technology adoption

What Skill Sprints are meant to achieve:

Skill Sprints are designed as a holistic approach to adopting new technology and transforming an organisation. Training is only concerned with individuals - with no attempt to transform the organisation. Consultants are typically disinterested in training. **Technology adoption and Digital Transformation require a process that takes care of the individual and organisational journey. Skil Sprints are designed to fill this gap.**

Skill Sprint is both a concept and an applied methodology: a new approach to technology adoption for companies that understand that staff development is a powerful tool for competitive advantage.

To upskill individuals and enable sustainable change, Skill Sprints are designed to guide the participants to a specific outcome. In order for a transformation to be successful it has to be driven by members of the organisation. Skill Sprints are designed to provide a framework and supportive environment for a small sustainable organisational change.

The participants must be the agents of change. The Skill Sprint captain supports and enables the participants but is not directly part of the change process. The captain is responsible for ensuring that a Liminal Experience is enjoyed by the participants to allow them to emotionally connect and transform with the change.

They provide liminal experience by creating an emotional connection between the participants and their tools for the onboarding of the new technology so that they be prepared and invested should any transformation of their work take place.

Skill Sprints provide an ideal learning environment for skills acquisition, retention and application. Allowing technical and non-technical participants to work together on a project allows them to address specific and immediate issues (e.g. infrastructure complexity). It also allows team members to maximise their skills and reduces technical debt.

Led by an expert in a real world environment, a Skill Sprint emphasizes the proven benefits of high quality interactions to facilitate ownership of skills, artefacts and knowledge sharing.

Where, What and How

A Skill Sprint typically consists of three participants: two participants from the target organization’s plus a subject matter expert called a *captain*. The Skill Sprint captain provides best practices and patternswhile the participants code,build and learn. The Captain is not allowed to touch the keyboard but consistently encourages quality interactions within the small team, and ensures that the participants switch roles regularly. This enables efficient methods of communicating information within the team through face-to-face conversation.

Before the Skill Sprint, the participants, taking input from any stakeholders, outline a goal that is specific to their business and that delivers real value. Because they create it themselves, they also own it, and are able to improve on it and share associated knowledge after the Skill Sprint is completed. After completing a Skill Sprint, they return to their original teams and, bring the newly acquired expertise to their colleagues.. Members of the team who did not participate in the Skill Sprint interact with, and learn through, the development process.

In summary: Knowledge sharing and integration through social interactions, repetition and ownership make the Skill Sprint an extremely viable and promising method for technology adoption on development teams. Constant reviewing consolidates the knowledge and creates an environment where participants are able to overcome the forgetting curve.

Skill Sprints are a one week process 1-week syllabus **customized** for the specific problem at hand. With a strong focus on developer empowerment and team autonomy, this paper presents the methodology as a solution to the challenges that come with technology adoption and digital transformation.

Skill Sprint is a combination of technical consulting with instructional design. Its systematic approach to learning creates objectives to reach a viable and deliverable outcome. *In summary, the goal of the Sprint is to ensure that the teams are properly engaged during* and *after the process.*

The Sprint is customised to suit the individual client objective. The customisation process will be based on the following observations:

- Figuring out the learning objective
- Inferring changes at the organisation level
- Whether the participants require further support after the Sprint has ended
- Analysing the real business value delivered after the Sprint

The Skill Sprint can also be used for learning new technology, including coding bootcamp and DevOps.

What to expect from a Skill Sprint:

Participants enter into the Skill Sprint with the expectation of creating a real valuable outcome: a product, a structural change or both. The majority of Skill Sprint deliveries address the same type of issues, however it's also important to focus on the remaining percentage of customization that is unique to each customer. Every Skill Sprint is preceded by a preparation process that includes:

- Researching and understanding of the problems unique to the customer
- Creation of a Skill Sprint syllabus, a rough guide for the technical concepts
- Testing of various service providers and preparation of a tailor-made technical solution;
- Selection and preperation of theparticipants.Those attending will become agents of change within their organizations. As such, they should have the appropriate interpersonal skills and motivation to enable these changes

The Sprint will require the Captain to work directly with other participants in a private space, full time a week. The Captain will teach concepts without touching a keyboard, and attendees will implement these concepts immediately. The Captain will continuously guide and motivate the participants, changing course when necessary to keep participants on track. A final result will be delivered, built entirely by the participants.

After the Sprint, participants will return to their development teams, where they will re-assimilate and exchange both technical knowledge and the final product. They will immediately be challenged to further develop and support the product. ,

Core concepts behind Skill Sprints

1. Tacit Knowledge

Use of technology relies on employees' expert knowledge. This kind of expert knowledge is mostly tacit knowledge, as opposed to formal or explicit knowledge. *Michael Polanyi, the noted Polymath first described tacit knowledge as knowledge that cannot be articulated. In other words, it is difficult to transfer by means of writing it down or verbalising it.*

Those experiences are attained through socialisation where mentoring is key. Studies have found a positive correlation between knowledge transfer through socialisation and better team performance and completion of tasks.[2]

The interdependent nature of labor in software development teams makes tacit knowledge an essential component of team performance. Companies should leverage this practice in their strategies of technology adoption. *A 2013 study conducted by Ryan and O'Connor found that teams with effective and functional tacit knowledge have a competitive advantage in developing new products and bringing them to market*.[3]

[2] Internal social networks in work teams: Structure, knowledge sharing and performance. Kaisa Henttonen, Minna Janhonen and Jan-Erik Johanson, 2013

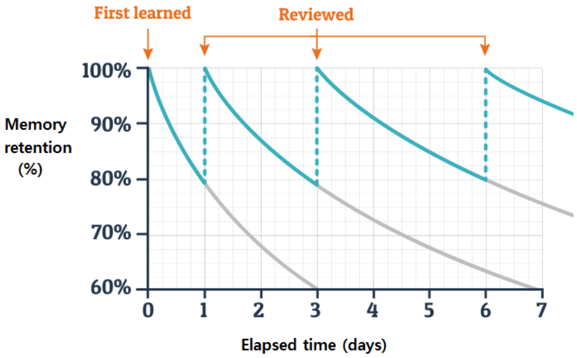
[3] Acquiring and Sharing Tacit Knowledge in Software Development Teams: An Empirical Study, Ryan & O'Connor, 2013

2. Agile

“Agile” software development allows an organisation to develop a sustainable relationship with technologies. Previous to agile, technology was developed in projects with a defined start and end. Today, our relationship with tech is continuous and ever changing. Agile is a set of principles and values that facilitates a healthy relationship between organisations and technology.

3. The Forgetting Curve

The forgetting curve hypothesizes the decline of memory retention in time. This curve shows how information is lost over time when there is no attempt to retain it.



4. Transactive Memory Systems

Transactive memory systems help to explain the dissemination and use of expert tacit knowledge in teams. TMS is a shared mental model that describes a cooperative division of labour for learning, remembering and communicating relevant knowledge.[4] The system includes transactive memory, meaning the individual memories and their interactions. *Simply put, it is understanding who or how to ask for information from others, communicating it effectively and using it in collective decisions.*[5][6]

[4] Effect of transactive memory systems on team performance mediated by knowledge transfer , Wang, Huang, Davison, and Yang, 2018

[5] Cognitive interdependence and convergent expectation in transactive memory. Andrea B Hollingshead, 2002

[6] Sharing of Tacit Knowledge in Organizations. Haradhan Kumar Mohajan, 2016

5. Pair Programming

Pairing is During this process the participants, regularly switch between between*navigating* and *driveing* – team up to code using one workstation.

With the guidance and knowledge of a navigator, the driver absorbs new knowledge by building on their own. The participants are able to work in a controlled environment with a learning-by-doing structure, which facilitates tacit knowledge exchange.

6. Liminal Experience

In anthropology, liminality (from the Latin word līmen, meaning "a threshold") is the quality of ambiguity or disorientation that occurs in the middle stage of a rite of passage, when participants no longer hold their pre-ritual status but have not yet begun the transition to the status they will hold when the rite is complete. During a rite's liminal stage, participants "stand at the threshold" between their previous way of structuring their identity, time, or community, and a new way, which completing the rite establishes.

Limitations of a Skill Sprint

The Skill Sprint was designed to ensure that all the workflow and the lifecycle of the technology remains within the organisation, delivered by existing employees and not external consultants so that it can continue indefinitely. The main limitation of Skill Sprints is the existence of *prior knowledge within the participants*.

Skill Sprints are intended to help skilled, experienced employees effect change within their organisations. Skill sprints provide a framework for holistic change therefore those involved need to deeply understand the nature of the Business.

Teams who had existing knowledge on technology were able to quickly adapt the lessons delivered through Skill Sprint and apply them to the challenges without relying on external help. Teams who had no prior technical skills or knowledge had difficulties understanding the scope of the Sprint itself. *Skill Sprints are unlikely to be successful if the participants do not have the prior knowledge and understanding of the problem itself*.

References

- Managing technical debt: Reducing friction in software development, Philippe Kruchten Robert Nord and Ipek Ozkaya, 2019
- Acquiring and Sharing Tacit Knowledge in Software Development Teams: An Empirical Study, Ryan & O’Connor, 2013
- Internal social networks in work teams: Structure, knowledge sharing and performance. Kaisa Henttonen, Minna Janhonen and Jan-Erik Johanson, 2013
- Effect of transactive memory systems on team performance mediated by knowledge transfer , Wang, Huang, Davison, and Yang, 2018
- Tacit knowledge and visual expertise in medical diagnostic reasoning: Implications for medical education. Peter Johan Heiberg Engel, 2009
- Managing Tacit Knowledge for a Software Development Process: A Case Study. David P. Stevens, Sonya HY Hsu and Zhiwei Zhu, 2012
- Applied Knowledge Retention – Are Active Learning Tools the Solution? Sushil Acharya, Bruce R Maxim and Jeffrey Jonathan Yackley, 2019
- Audiovisual methods in teaching. Edgar Dale, 1969
- Learning and Forgetting Curves in Software Development: Does Type of Knowledge Matter? Keumseok Kang and Jungpil Hahn, 2009
- Learning and Forgetting Models and Their Applications. MY Jaber, 2006
- The concept of "Ba": Building a foundation for knowledge creation. Ikujiro Nonaka and Noboru Konno, 1998
- Three ‘C’s of Agile Practice: Collaboration, Co-ordination and Communication. Helen Sharp and Hugh Robinson, 2010
- Cognitive interdependence and convergent expectation in transactive memory. Andrea B Hollingshead, 2002
- Sharing of Tacit Knowledge in Organizations. Haradhan Kumar Mohajan, 2016
- The Impact of Information Technology and Transactive Memory Systems on Knowledge Sharing, Application, and Team Performance: A Field Study. Sue Young Choi, Heeseok Lee, Youngjin Yoo, 2010

- Transactive Memory Systems: Current Issues and Future Research Directions. Kyle Lewis and Benjamin Herndon, 2011
- Strengthening the Case for Pair Programming. Laurie Williams, Robert R Kessler, Ward Cinningham and Ron Jeffries, 2000
- Mob programming – full team, full throttle, Narcus Hammarberg 2013
- Exploring alternative ways of assessing prior knowledge, its components and their relation to student achievement: a mathematics-based case study. Hailikari T, Nevgi A and Lindblom-Ylänne S, 2007
- Acquiring and teaching cognitive skills: a state-of-the-art of theory and research. DE Corte E, 1990
- The Relevance of Prior Knowledge in Learning and Instructional Design. Telle Hailikari, Nina Katajavuori, and Sari Lindblom-Ylanne, 2008