



Semester 2 Alternative Assessment 2019/20
End-of-Module Assignment

MS802: IS Strategy & Innovation (2019-20)

Student Name:	Jayakarthis Boovendran
Student ID:	19230487
Course:	MBY – MSc Business Analytics
Email:	J.Boovendran1@nuigalway.ie
Assignment Topic:	Innovation Assignment
Lecturers:	Dr Lorraine Morgan , Dr Denis Dennehy

Declaration

"In submitting this work, I confirm that it is entirely my own. I acknowledge that I may be invited to undertake an online interview if there is any concern in relation to the integrity of my submission."

Signature:

Date: 12 May 2020

A handwritten signature in black ink, appearing to read "Jayakarthis".

Jayakarthis Boovendran

Section – A

Q2. The InnerSource Approach to Software Development & Innovation

I. Introduction

With technology-advancements comes the risk of picking the appropriate strategies for software development. Ever since the late '90s, Open source communities have been highly successful in delivering several notable software products. Inspired by their success, many other giant corporations have started to investigate and implement open-source development strategies into their internal software development practices to attain a similar degree of success. This adaptation of practising open-source strategies internally within a company is called the InnerSource (Stol et al. 2015).

II. InnerSource – The Open Source Way

InnerSource comprises of diverse teams of stakeholders, developers and business units collaborating and contributing to the In-house software development. The key factors that define InnerSource are transparency, openness to contributions and improvement through collaboration.

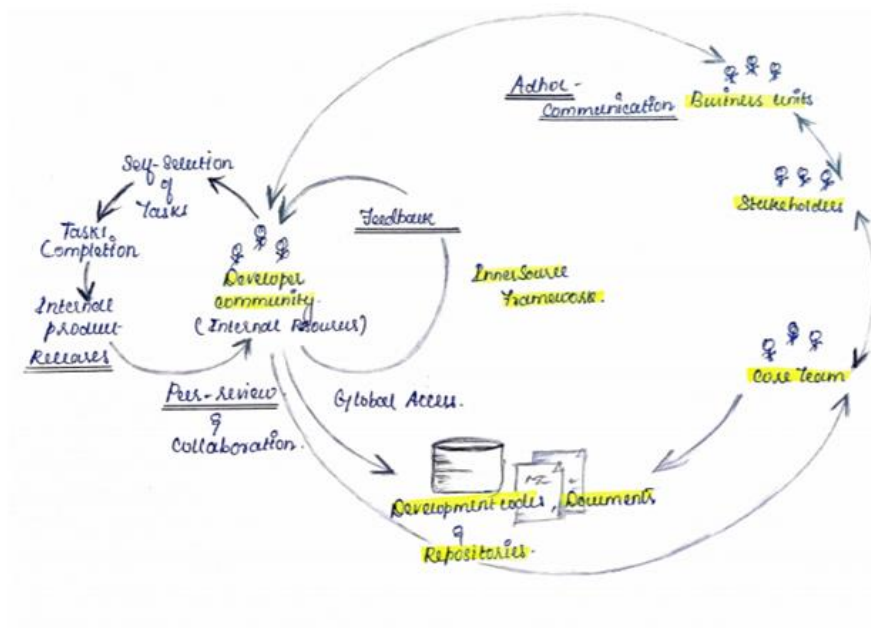


Figure 1: Actors & Key Processes of InnerSource Framework

III. Why InnerSource?

There is a wide-spread interest in InnerSource practices among more giant corporations as they provide the core benefits of Open Source development. A few are:

- **Reusability of development artefacts & Accelerated development time:** Allowing full-access to development codes and documents encourages the developers to reuse them appropriately, thus, saves time and effort (Stol et al. 2014).
- **Improved software quality:** Collaborative effort in development and consistent quality assurance through peer-review assures better software quality
- **Open to innovation:** According to (David Rock, 2016), the participation of diverse teams in software development produces more useful outcomes in an environment like the InnerSource.
- **Opportunities for personal development** through working in projects with different architectures, tech-stack and technologies (Stol et al. 2014)

IV. Adopting to InnerSource

Though the principles of inner source inherit from open source development, there are no defined methodologies. Many companies who successfully implemented InnerSource followed their approach. However, Stol et al. (2015) defined a set of critical factors for adopting InnerSource. These guidelines are results of extensive research and observations from a subjects-matter interview conducted by the author.

a) Product Suitability

- **Seed Products:** As an initial step, to attract the developer community, the organization should select a seed product with the existing initial implementation. Furthermore, the seed product should have sufficient functionality to work on & be based on a familiar tech stack and architecture.

- **The Stakeholders:** The selected Seed product needs to have stakeholders of different expertise. Leveraging the project to the organizational level, results in a variety of contributions and requirements from different product groups or business units, introducing more diversity in the contributions.
- **Modularity:** Modularity is the key to parallel development. It let developers self-select a subset of the functionality to which they can then contribute effectively. Furthermore, to ensure module cohesiveness, the complexity of the functional modules should be kept as low as possible.

b) Practices and Tools

- **Development Practices:** In order to accommodate new ideas regularly, the organization must follow a flexible approach rather than contemporary approaches like agile development, this, enables the developers to convert the ideas to working codes for peer-reviews quickly.
- **Quality Assurance:** The primary way of assuring quality is through Peer review of contributions by the developer community. Moreover, the team performs frequent product releases to facilitate consistent feedback from peers.
- **Development Tools:** The critical aspect of InnerSource development is attracting the developer community. Introducing new tools has risks of low productivity and delayed development time, also, may demotivate developers. Therefore, for better results, it is recommended to use development tools & techniques that the developers are already familiar with.

c) People & Management

- **Coordination and leadership:** The model requires a more flexible approach to coordinating resources. In this case, a core team is responsible for integrating and managing contributions—usually, the team comprised of developers of varied expertise who contributes significantly to the project. The critical responsibility here is to maintain the contributor's interest; however, the critical aspect of success in self-organization.
- **Transparency:** It is essential to maintain transparency across the corporation through the provision of infrastructures like full-access to development artefacts, legacy projects and access to repositories and a medium for knowledge sharing & internal communication.
- **Management Support & Motivation:** Management support is the key to the success of the inner Source model. Though it is hard to identify and estimate real progress, the management team plays a significant role in keeping the developers motivated and solving local issues.

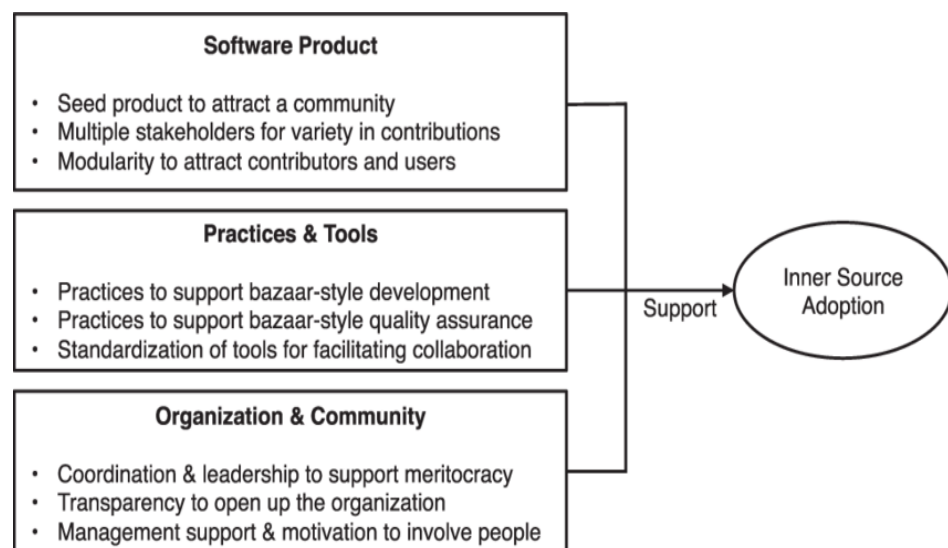


Figure 2: Actors & Key Processes of InnerSource Framework (Stol et al. 2014)

V. Case Study:

InnerSource Implementation @ Ericson: Ericson is a renowned Swedish multinational telecommunication industry. They adopted InnerSource to **transition from product-focused to platform-focused architecture**, because, the agile development principles insist on homogeneity rather than a wasteful effort on feature development (Cooper & Stol 2018).

Case Study : InnserSource Implementation @Ericson			
Category	Factor	Observation	
Product Suitability	Seed Product	Underlying platform	Ericson uses a layered architecture for its services. The objective is to transition from product-focused to a platform-focused architecture
	Stakeholders	Diverse Stakeholders	Since this is an architectural decision , stakeholders of diverse expertise were involved
	Modularity	Horizontal component of the layer	The Horizontal components were the potential bottlenecks while modifying an existing architecture.
Practices and Tools	Development Practices	Community Developed Software (CDS).	The Program Development Unit of Ericson created 'Community Developed Software' (CDS) to facilitate collaborative development among multiple teams using Open Source development practices.
	Quality Assurance	Trusted lieutenants, Linus's law	To ensure high quality,Trusted lieutenants rigorously reviewed the contributions before committing to the repository. Based on the famous Linus's law " Many eyeballs make all bugs shallow. "
	Development tools	Flexible CDS Infrastructure, Network Management Tools	Existing Development models and tools were used.
People and Management	Coordination & leadership	Core teams, Product owners, Architect, Code Guardian	Core teams - responsible for providing components to development teams, Product Owners - Responsible for product backlog management, Architech - Ensures the components are architecturally reliable, Code Guardian - Compiles with coding standards and rules
	Transparency	Community portal - for Internal Communication, Access to assets	To scale development capacity, any team within Ericson were allowed to access the development assets
	Management Support & Motivation	Governance Council	Involved a team of handpicked individuals who were known for their technical expertise and reputation. Inspired by Apache's Open Source Governance practices.

Figure 3: InnerSource Implementation at Ericson

VI. Conclusion

Adopting InnerSource does not guarantee overnight success. After all, it is a slow and consistent process that demands extensive initial research and defining proper success criteria.

References:

- Stol, K., & Avgeriou, P., & Ali, B. M., & Lucas, Yan & Fitzgerald, B. (2014), "Key *Factors for Adopting Inner Source*", ACM Transactions on Software Engineering and Methodology. 23. 10.1145/2533685.
- Stol, K., & Pär J. Å., & Brian F. (2015), "*Software Sourcing in the Age of Open: Leveraging the Unknown Workforce*", DOI: 10.1007/978-3-319-17266-8
- Carroll, N. (2017), "*How Is Value Created Within An Inner Source Environment?*", OpenSym '17, August 23–25, 2017, Galway, Ireland, Available at <https://doi.org/10.1145/3125433.3125453>
- Cooper, D. & Stol, K. (2018), "*Adopting InnerSource – Principles and Case studies*", USA: O'Reilly Media

Section – B

Q3. Design Thinking – A Foundation for Innovation?

I. Introduction

“You've got to start with the customer experience and work back toward the technology - not the other way around.”- Steve Jobs

In the modern era, businesses are striving to be more customer-centric to understand and respond to the changing customer expectations continually, accomplish longevity and competitive advantage in the market and to be innovative. This study revolves around one such practice that helps businesses achieve the before-mentioned benefits.

II. Design Thinking

Brown (2008) defines Design thinking as a problem-solving technique that offers innovative solutions to complicated problems by focusing primarily on customers.

III. Stages of Design Thinking

Platnner et al. (2011) proposed a five-stage design thinking process to understand the customers better. Besides, the proposed DT process model is iterative and non-linear.

a) Empathize

In order to make customer-centric decisions, it is critical to understand the problem from the customers' point of view. In this phase of design thinking, the business firm empathizes with the end-user to attain a more realistic view of the problem and identifies the customer expectations.

b) Define

In this phase, the team defines a clear problem statement alongside the success criteria based on the insights derived from the understanding. It is essential to accommodate the critical customer expectations in the problem statement.

c) Ideate

With the problem statement defined, in this phase, the design team look for innovative solutions or alternative ways to resolve the problem.

d) Prototype

In this phase, the team implements the proposed ideas by developing several inexpensive prototypes of the product/feature in order to identify and access the solution that best suits the problem.

e) Test

The team then evaluates the best-picked solution from the prototyping stage against the problem statement. It is a recommended practice to reiterate the stages of the DT process to redefine the problem further or refine the solution until the team obtains the optimum result to the problem

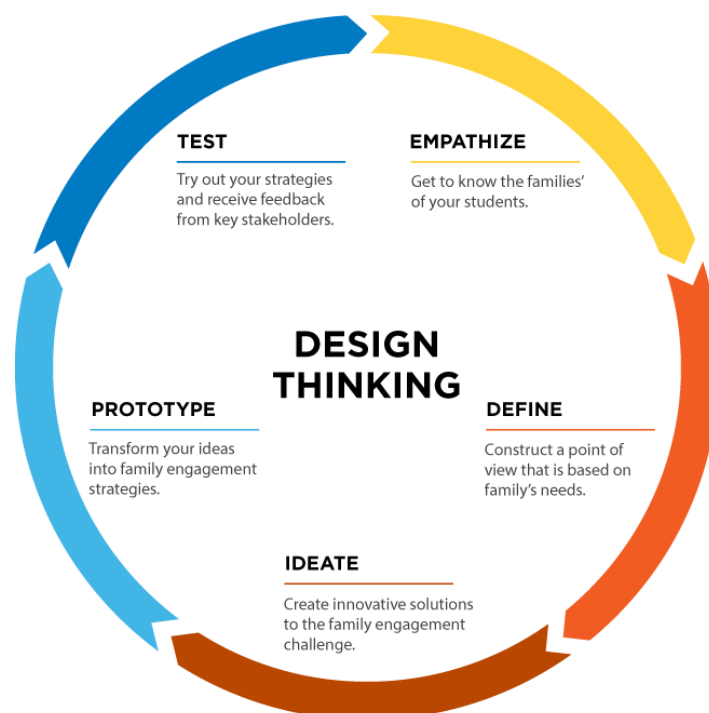


Figure 1: Five Stages of Design Thinking (raceacademy.com)

IV. DT Tools

DT tools play a significant role in expediting the design thinking process. These tools facilitate the design team in transforming abstract data into executable ideas. Tschimmel (2012) classifies the commonly used design thinking tools as follows.

a) Tools for understanding and Empathizing

Tools like **Mindmaps**, **Personas**, Empathy map, and Observation techniques like Self-documentation, photography, videos or Mobile Ethnography help capture behavioural patterns of customers or understand a situation in a systematic way. Furthermore, these tools pave the way for organizing complex information in a visual form that represents the perspectives of the end-users.

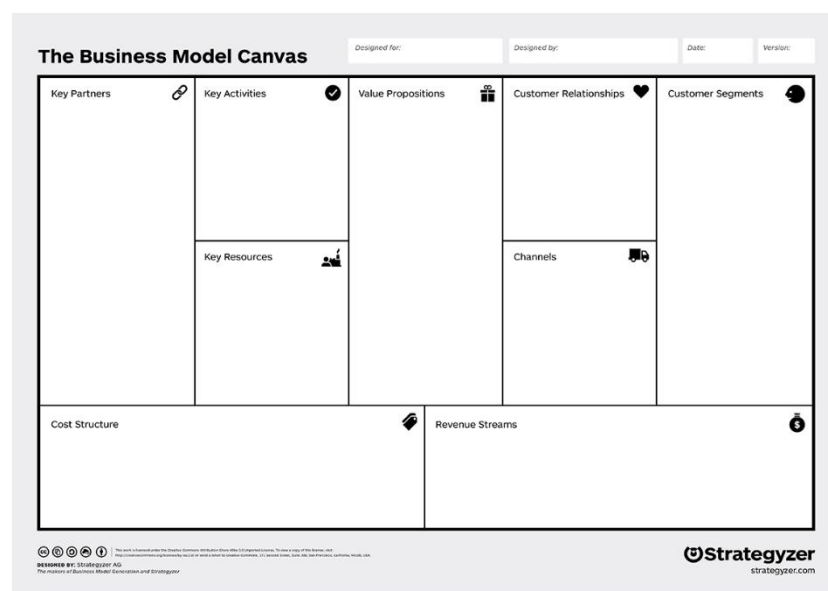


Figure 2: Business Model Canvas (www.strategyzer.com)

b) Tools for Idea Generation

These tools facilitate the generation of ideas and help visually representing them. DT tools like **Brainstorming**, Sketching and visual components like photographs, phrases, texts and images also play a significant role in transforming ideas into visual representations.

c) Tools for Development

These tools serve the purpose of elaboration and development of ideas. Commonly used DT tools are Storyboards, Technical drawings, Sketches, and **Rapid Prototyping**.

d) Tools for Communication

These tools are employed in communicating the results to the stakeholders, also, in improving the prototype through feedback. In real-time, **Storytelling** technique is employed for presenting the findings alongside Storyboard or Roleplay.

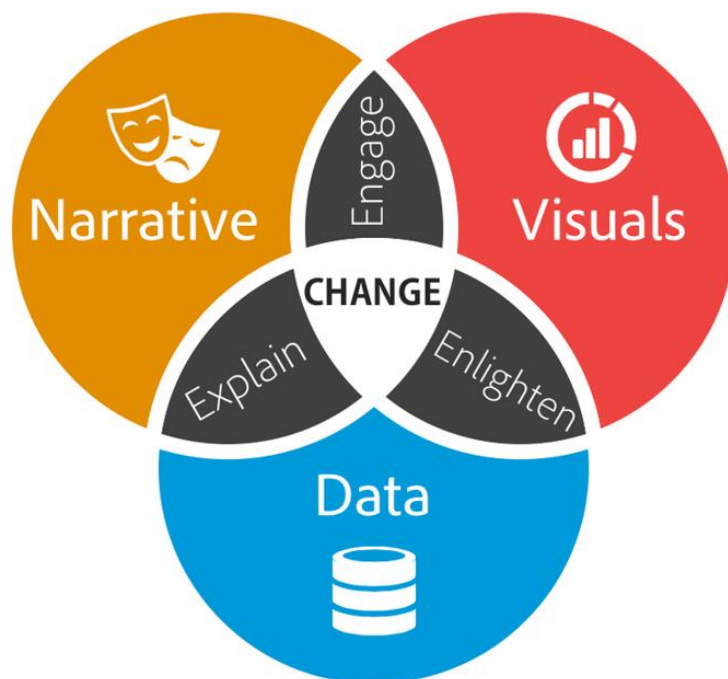


Figure 3: Key Components of Data Storytelling (www.forbes.com)

V. Challenges

DT offers positive outcomes on innovation management, but implementing DT in industrial settings is challenging. Based on interviews with five large industries with more than five years of experience working in DT, Carlgren et al. (2016), proposes the critical challenges of DT as follows.

1. **Misalignment with the existing process:** The iterative and non-linear nature of DT contradicts with the linear mainstream process. Furthermore, DT tasks are resource-intensive and are hard to achieve in geographically distributed teams and risk-averse culture.
2. DT results are **challenging to implement** because of time constraints, or they fall beyond the scope of the project.
3. **Difficult to appraise DT activities:** Measuring and evaluating the outcome and tracing the contribution of DT activities are difficult
4. Teams perceive DT as a questioning of the existing practices. Therefore, they are reluctant to learn DT concepts.
5. Requires **skills** relating to the use of design tools, data visualization and building rough prototypes, which are **hard to acquire**
6. There are **communication barriers** related to the principles of DT. Further, are challenges in converting information into executable technical requirements.

VI. Responsibilities of Innovation Manager

Managers are responsible for understanding the value of DT and evaluating whether the problem requires a DT approach. Further, the manager chooses the DT model that best suits the organizational practices. In addition to that, the manager should develop a well-defined road-map to guide the team throughout the DT process. Still, in most organizations, DT is a new initiative. Therefore, the managers should create a stimulating workplace and encourage the team to participate in the processes actively.

VII. Conclusion

Design Thinking is not a one-step solution to success. There are proofs enough to justify the fact that DT promotes innovation in an organization. However, many are lost in the process of DT or have had short-lived success. Brown (2015) argues that DT does not facilitate a competitive advantage if not implemented in a sustained way. Therefore, instead of implementing DT for a product or a feature, organizations should make it a routine practice to attain the full benefits.

Reference

- Leavy, B. (2010), "*Design thinking – a new mental model of value innovation*", *Strategy & Leadership*, Vol. 38 No. 3, pp. 5-14.
<https://doi.org/10.1108/10878571011042050>
- Tschimmel, K. (2012). "*Design Thinking as an effective Toolkit for Innovation*", *Proceedings of the XXIII ISPIM Conference: Action for Innovation: Innovating from Experience*. Barcelona. ISBN 978-952-265-243-0.
- Beckman, S. L. and Barry, M. (2007) "*Innovation as a Learning Process: Embedding Design Thinking*", *California Management Review*, 50(1), pp. 25–56. DOI: 10.2307/41166415.
- Carlgren, L., Elmquist, M., Rauth, I, (2016), "*The Challenges of Using Design Thinking in Industry – Experiences from Five Large Firms*",
<https://doi.org/10.1111/caim.12176>
- Brown, T (2015), "When Everyone Is Doing Design Thinking, Is It Still a Competitive Advantage?", Available at: <https://hbr.org/2015/08/when-everyone-is-doing-design-thinking-is-it-still-a-competitive-advantage> [Accessed 12 May 2020]
- Plattner, H., Leifer, L., Meinel C (2008), "*Design Thinking – Understand – Improve - Apply*", DOI 10.1007/978-3-642-13757-0, Available at <http://www.springer.com/series/8802> [Accessed 12 May 2020]