# Emerging Technologies for Innovation Management in the Software Industry


Varun Gupta
*Universidad de Alcalá, Madrid, Spain*

Chetna Gupta
*Jaypee Institute of Information Technology, Noida, India*



A volume in the Advances in Systems Analysis, Software Engineering, and High Performance Computing (ASASEHPC) Book Series

Published in the United States of America by
IGI Global
Engineering Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA, USA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com



Library of Congress Cataloging-in-Publication Data

Names: Gupta, Varun, 1987- editor. | Gupta, Chetna, editor.
Title: Emerging technologies for innovation management in the software industry / Varun Gupta, and Chetna Gupta, editors.
Description: Hershey, PA : Business Science Reference, [2022] | Includes bibliographical references and index. | Summary: "This book will highlight the role of technology to assist software companies, especially small start-ups, to innovate their products, processes or business models"-- Provided by publisher.
Identifiers: LCCN 2021042348 (print) | LCCN 2021042349 (ebook) | ISBN 9781799890591 (hardcover) | ISBN 9781799890607 (paperback) | ISBN 9781799890614 (ebook)
Subjects: LCSH: Computer software industry--Management. | Technological innovations. | Computer software--Development.
Classification: LCC HD9696.63.A2 E55 2022 (print) | LCC HD9696.63.A2 (ebook) | DDC 005.068/4--dc23/eng/20211014
LC record available at https://lccn.loc.gov/2021042348
LC ebook record available at https://lccn.loc.gov/2021042349

This book is published in the IGI Global book series Advances in Systems Analysis, Software Engineering, and High Performance Computing (ASASEHPC) (ISSN: 2327-3453; eISSN: 2327-3461)

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

# Advances in Systems Analysis, Software Engineering, and High Performance Computing (ASASEHPC) Book Series

Vijayan Sugumaran
Oakland University, USA

ISSN:2327-3453
EISSN:2327-3461

## Mission

The theory and practice of computing applications and distributed systems has emerged as one of the key areas of research driving innovations in business, engineering, and science. The fields of software engineering, systems analysis, and high performance computing offer a wide range of applications and solutions in solving computational problems for any modern organization.

The **Advances in Systems Analysis, Software Engineering, and High Performance Computing (ASASEHPC) Book Series** brings together research in the areas of distributed computing, systems and software engineering, high performance computing, and service science. This collection of publications is useful for academics, researchers, and practitioners seeking the latest practices and knowledge in this field.

## Coverage

- Computer Graphics
- Performance Modelling
- Metadata and Semantic Web
- Human-Computer Interaction
- Computer System Analysis
- Computer Networking
- Engineering Environments
- Storage Systems
- Network Management
- Enterprise Information Systems

IGI Global is currently accepting manuscripts for publication within this series. To submit a proposal for a volume in this series, please contact our Acquisition Editors at Acquisitions@igi-global.com or visit: http://www.igi-global.com/publish/.

The Advances in Systems Analysis, Software Engineering, and High Performance Computing (ASASEHPC) Book Series (ISSN 2327-3453) is published by IGI Global, 701 E. Chocolate Avenue, Hershey, PA 17033-1240, USA, www.igi-global.com. This series is composed of titles available for purchase individually; each title is edited to be contextually exclusive from any other title within the series. For pricing and ordering information please visit http://www.igi-global.com/book-series/advances-systems-analysis-software-engineering/73689. Postmaster: Send all address changes to above address. 

## Titles in this Series

*For a list of additional titles in this series, please visit: http://www.igi-global.com/book-series/advances-systems-analysis-software-engineering/73689*

***Technology Road Mapping for Quantum Computing and Engineering***
Brojo Kishore Mishra (GIET University, India)
Engineering Science Reference • © 2022 • 243pp • H/C (ISBN: 9781799891833) • US $225.00

***Designing User Interfaces With a Data Science Approach***
Abhijit Narayanrao Banubakode (MET Institute of Computer Science, India) Ganesh Dattatray Bhutkar (Vishwakarma Institute of Technology, India) Yohannes Kurniawan (Bina Nusantara University, Indonesia) and Chhaya Santosh Gosavi (MKSSS's Cummins College of Engineering, India)
Engineering Science Reference • © 2022 • 325pp • H/C (ISBN: 9781799891215) • US $245.00

***Implementation of Machine Learning Algorithms Using Control-Flow and Dataflow Paradigms***
Veljko Milutinović (Indiana University, Bloomington, USA) Nenad Mitić (University of Belgrade, Serbia) Aleksandar Kartelj (University of Belgrade, Serbia) and Miloš Kotlar (University of Belgrade, Serbia)
Engineering Science Reference • © 2022 • 296pp • H/C (ISBN: 9781799883500) • US $245.00

***Advancing Smarter and More Secure Industrial Applications Using AI, IoT, and Blockchain Technology***
Kavita Saini (Galgotias University, India) and Pethuru Raj (Reliance Jio Platforms Ltd., Bangalore, India)
Engineering Science Reference • © 2022 • 309pp • H/C (ISBN: 9781799883678) • US $245.00

***Deep Learning Applications for Cyber-Physical Systems***
Monica R. Mundada (M.S. Ramaiah Institute of Technology, India) S. Seema (M.S. Ramaiah Institute of Technology, India) Srinivasa K.G. (National Institute of Technical Teachers Training and Research, Chandigarh, India) and M. Shilpa (M.S. Ramaiah Institute of Technology, India)
Engineering Science Reference • © 2022 • 293pp • H/C (ISBN: 9781799881612) • US $245.00

***Design, Applications, and Maintenance of Cyber-Physical Systems***
Pierluigi Rea (University of Cagliari, Italy) Erika Ottaviano (University of Cassino and Southern Lazio, Italy) José Machado (University of Minho, Portugal) and Katarzyna Antosz (Rzeszow University of Technology, Poland)
Engineering Science Reference • © 2021 • 314pp • H/C (ISBN: 9781799867210) • US $225.00

***Methodologies and Applications of Computational Statistics for Machine Intelligence***
Debabrata Samanta (Christ University (Deemed), India) Raghavendra Rao Althar (QMS, First American India, Bangalore, India) Sabyasachi Pramanik (Haldia Institute of Technology, India) and Soumi Dutta (Institute of Engineering and Management, Kolkata, India)
Engineering Science Reference • © 2021 • 277pp • H/C (ISBN: 9781799877011) • US $245.00

IGI Global
PUBLISHER of TIMELY KNOWLEDGE

701 East Chocolate Avenue, Hershey, PA 17033, USA
Tel: 717-533-8845 x100 • Fax: 717-533-8661
E-Mail: cust@igi-global.com • www.igi-global.com

# Preface

## AN OVERVIEW

Innovation is the key to maintain competitive advantage. Innovation in products, processes, and business models help companies to provide economic value to their customers. Identifying the innovative ideas, implementing those ideas, and absorbing them in the market requires investing many resources that could incur large costs. Technology encourages companies to foster innovation to remain competitive in the marketplace.

*Emerging Technologies for Innovation Management in the Software Industry* serves as a resource for technology absorption in companies supporting innovation. It highlights the role of technology to assist software companies—especially small start-ups—to innovate their products, processes, and business models. This book provides the necessary guidelines of which tools to use and under what situations. Covering topics such as risk management, prioritization approaches, and digitally-enabled innovation processes, this premier reference source is an ideal resource for entrepreneurs, software developers, software managers, business leaders, engineers, students and faculty of higher education, researchers, and academicians.

## TARGET AUDIENCE

Entrepreneurs, Software Engineers, Scientists working as Researchers with research organizations, Universities, and Research & Development Units, Academicians, Business consultants, and others will benefit from the research findings presented in this book. This book gives readers a single point of entry to research on startup innovation in terms of empirical investigations and research solutions. The book's information provides significant direction to the startup community and other stakeholders for incorporating it into their actual business processes. The research papers presented in this book will assist the audience in the following ways:

### Entrepreneurs

- Improving their current innovation strategies.
- Using the knowledge imparted in this book to solve their current business difficulties.
- Developing dynamic capacities to maintain a market competitive advantage.

### Academicians

- Lecturing students on current innovation management strategies used by startups.
- Developing new research lines and innovating existing research lines using the book.
- Creating strong research proposals that can be submitted to funding agencies.

### Scientists/Researchers

- Using the book as a source of information to start new research projects or improve on existing ones.
- Developing strong research project proposals for possible funding agency submissions.
- Strengthening their ties with industry and addressing real-world issues.
- Using a book as a main research study to find knowledge gaps and acquire ideas for future research in their professional and academic fields.

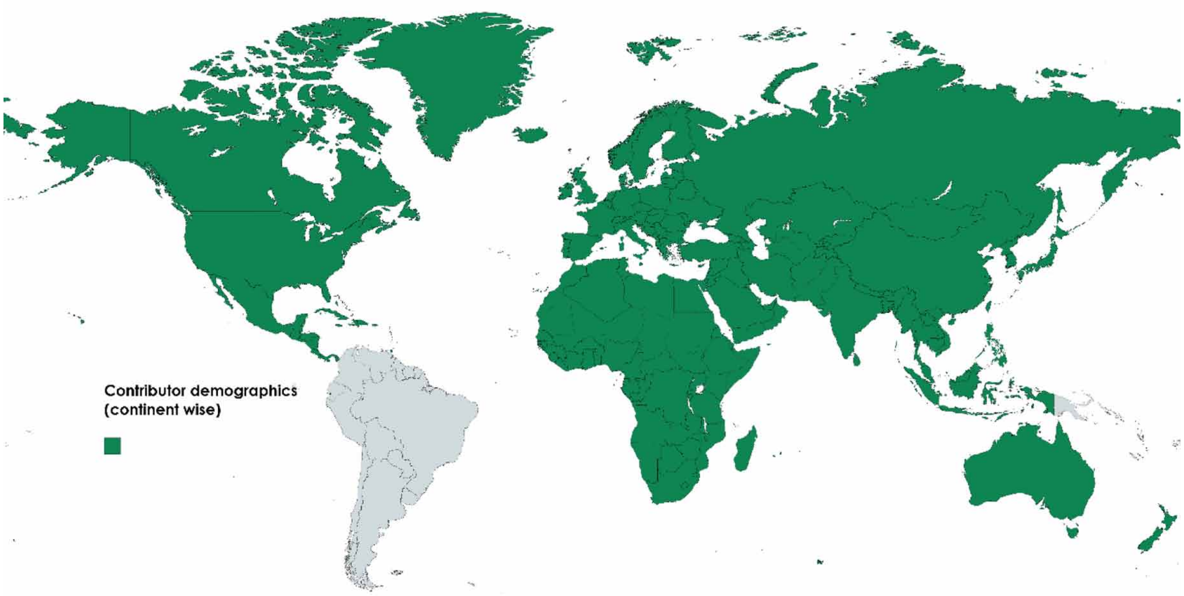
### Software Engineers

- Improving startup business operations, for example, by making them better and more suited to startup working environments, based on knowledge imparted in the book.
- Incorporating the findings presented in the book into their daily work routines.

## CONTRIBUTOR DEMOGRAPHICS

This book includes contributions from 39 authors from prestigious universities throughout the world. These authors contributed to 14 of the book's chapters. Contributors came from all over the world, including North America (United States of America (USA)), Europe (Denmark, Norway Portugal, Russia, Spain, Turkey, United Kingdom (UK)), Australia (Australia), Asia (India, Pakistan, Vietnam, Malaysia) and Africa (South Africa) as graphically represented in Figure 1).

*Figure 1. Contributor demographics (based on their affiliations)*




## ORGANISATION OF BOOK

This book is divided into 14 chapters, each of which focuses on innovation management in the context of a startup. These investigations are organized into 14 chapters, each of which is detailed in detail in the following lines.

Chapter 1 highlighted that Software startups have been widely known for their potential for disruptive innovation and their ability to generate wealth through unique value propositions and business models. The benefits that such organizations provide to the local and global economy are well documented. There is however that a concern when it comes to software startups is that most such startups fail in less than 2 years of inception. Given the invaluable contributions that these emerging organizations bring to the lives of their founding teams and the overall economic system alike, the causes, the current constructs contributing to the failure and possible success of software startups should merit further study.

From the perspective of business model development, much of the software startup space is presently dominated by the agile paradigm of business model creation using methodologies such as the Business Model Canvas, the Lean Startup and the Lean Canvas. The traditional business model development methodology has largely been abandoned given that it takes too much time to work through and the ever-changing fast paced nature of the software market would need business modelling methods which can easily pivot and is quick to develop, form hypothesis and test. The situation is not very different in the product development space, as the agile manifesto was originally conceived with the needs of the software space in mind. User stories are one of the most popular methodologies used when capturing requirements and prioritization is done using techniques like QFD, pair-wise analysis and MoSCoW analysis. Prototyping is used as a tool to help with testing out product concepts and getting user feedback. Some of the shortcomings observed during the creation of the startup is the lack of a holistic approach to the startup's development. The product and business model development efforts are rather created in silos with little interconnections. For instance, there is no commercial validation efforts done during the product development process. All efforts during that point is made to ensure that the product being developed is useful to the user and has good usability. There are similar issues when it comes to business model development. A better paradigm could be to explore an integrated approach for development of the startup, where both product and business model development are explored as a part of the same process with a common goal in mind. Ideally, a framework developed using such a perspective will incorporate additional aspects such as making use of product evangelists to promote the value proposition, keeping a close watch on the evolutions within the user's problem space, creating potential product ecosystems around the core value proposition and leveraging metrics for better decision making.

Chapter 2 highlighted that the widespread use of digital technology in innovation processes and outcomes, has prompted scholars to develop new theories on innovation management. These theories challenge long-held beliefs about the relationship between company performance and innovation processes, as well as the boundaries between innovation and organizations. Researchers must study and investigate digital technology implementation in order to foster innovative activity, which necessitates new digital technology theorization. Scholars have developed various research directions to theoretically understand digital technology in relation to developing digital business strategies, reassembling current capabilities with digital resources to establish digital capabilities, and capturing or creating value using digital technologies.

A thorough understanding of how digital technology affects the management of the innovation process might lead to the creation of an innovation process framework. Based on these premises and the research streams mentioned, researchers argue that incorporating and using digital technologies in innovation processes forces organizations to reorganize their business models and manage the innovation process in a different way than previously stated in the literature. The mechanisms that digital innovation supports are sometimes forgotten in the story of digital innovation. The potential of digital innovation to reconfigure, revitalize, challenge, and rethink the way things are viewed and comprehended is its primary impact. To put it another way, digital innovation is all about what it changes and how it affects how things are done as a result of the use of digital technologies. To comprehend change, one must first comprehend the mechanism through which change occurs, and vice versa. As a result, in the context of digital innovation, business process management is becoming increasingly important.

Digital innovation is expected to alter the process by balancing new innovation features with immediate feedback, balancing adaptation freedom with predefined structure, balancing positive deviation with process compliance, and balancing inter-organizational emergence with intra-organizational optimisation, according to researchers' attempts to link digital innovation to BPM. A new stream of research has recently emerged at the interface of digital innovation and business process management. This stream presents new ideas to describe how digital innovation affects the design, analysis, and management of business processes by enabling, hindering, shifting, or constraining them. It also looks at how BPM theory, technology, and practices can help us understand the processes and outcomes of digital innovation. The goal of this stream is to bring together those two disparate and isolated fields so that their insights, ideas, and theories can collide and transcend the bounds of their own literature streams.

Despite the fact that the literature on BPM and digital innovation is fragmented, it is apparent that the two sectors are complementary and mutually beneficial. Scholars in both fields must examine their techniques, questions, and assumptions in order to assess this complementarity. Scholars must start listening in on each other's talks in order to contribute to this complementarity. Context comprehension is a significant source of research prospects in both digital innovation and business process management. Contextual factors have a significant impact on both digital innovation and business process management.

The BPM field has produced context-aware methodologies, tools, and conceptualizations. To address context, the digital innovation area has used computational and empirical methodologies. The possibility arises from using digital innovation research methodology to the development and improvement of BPM technology, such as process analysis and process mining, and vice versa.

Chapter 3 introduced an area of research related to the implementation of emerging Unified Communications and Collaboration (UC&C) technologies for productivity and innovation management within the context of large-scale automotive design, manufacture and business operations at General Motors (GM), a leader in the global automotive industry. It further discusses how the chapter bridges the gaps presented through the design of the research developed with the purpose of evaluating the impact of said emerging technologies. In terms of mentioning what problems existed, prior to the research undertaking reported on in this chapter, General Motors had not implemented unified communications within its manufacturing, design or business operation functions and had not engaged in the development of an Internet of Things (IoT)-related digitization strategy.

Chapter 4 highlighted that the hardware startups are increasingly popular due to recent advancements in hardware technologies. Nowadays, hardware product development involves the process innovation not only at the hardware level but also at software components. The scarce of knowledge on hardware startup product development motivates us to carry an empirical investigation on five hardware startup companies. This chapter reported some common good practices among hardware startups, i.e., process definition, evolutionary development process and document management. Several factors that are different from software startups, such as low priority of product quality, product pipeline and unrecognized product platform, are revealed. An integrative process model of hardware product development that shows the connections between human factors in the startups, their speed-prioritized development processes, and the consequence of hindered productivity in the later phases, is finally proposed. The model has some implications for hardware startup founders to plan for the trade-off between team, speed, quality, and later productivity.

Chapter 5 highlighted that in highly dynamic situations, entrepreneurs build value propositions in resource-constrained conditions. The activity is set up as a series of experiments, with each one aimed at validating value proposition-related assumptions with customers. Validation entails interactions between potential customers and the startup team utilizing prototypes, which leads to the confirmation of current assumptions as well as the discovery of new insights that lead to more experiments. The main features of the value proposition identification model are highlighted, and a novel value prioritizing approach is proposed in this chapter.

Chapter 6 highlighted that in today's world of strong competition, firms aim to minimize costs as much as possible while enhancing efficiency and quality. As a result, in order to keep the company afloat, managers must manage multiple crises at the same time. Organizations are putting a larger focus on concepts like novelty, creativity, and speed as a result of the rapid rise of technology and globalization, which allows knowledge to become a strategic value. To manage turbulence while maintaining long-term survival, businesses must participate in innovative operations. The notions that are currently separating companies are knowledge and the human factor that creates knowledge. The development, sharing, application, and management of knowledge are all elements of the organizational culture. The relevance of knowledge management is demonstrated by the fact that knowledge may be exploited as a competitive weapon by businesses in a global society, and that capital-intensive enterprises are being replaced by knowledge-intensive businesses. Organizations, on the other hand, may encounter a knowledge gap between their existing level of knowledge and the amount of knowledge required to create new items, processes, or services. Businesses will need to conduct new research to close this gap. The most important factors for a corporation at this time are the quality and quantity of knowledge, as well as how it will be employed to carry out innovative operations. Integration of knowledge management and knowledge processes boosts a company's innovation and performance.

Knowledge management strategies and initiative actions used by businesses include knowledge development, transmission, application, and storage. A company's ability to innovate is boosted by effective knowledge management. Knowledge development in enterprises serves as a foundation for innovation and competition. As a result, companies can use knowledge management to help them launch new goods and services. In knowledge-based economies, knowledge is a critical resource for businesses to develop effective management policies and practices. Thanks to knowledge management, businesses improve their ability to innovate, increase productivity, and, as a result, gain a competitive advantage in the medium to long term. Firms aim to extend their innovation activities and produce value with their growing expertise. In terms of businesses, knowledge management is at the heart of the innovation process and organizational harmony.

Businesses should focus on a variety of activities that allow them to both follow and develop innovations while also taking advantage of available capabilities since exploratory and exploratory activities compete for precious resources. The purpose of this chapter was to investigate the impact of ambidexterity on company success via knowledge management and innovation. As a result of the analysis within the context of a designated model, we concluded that there is a positive and significant effect of knowledge management over innovation, a positive and significant effect of innovation over ambidexterity, and a positive and significant effect of ambidexterity over business performance. After using the structural equation modeling linearity hypothesis to run the model, it became clear that seeking ambidexterity through inventive activities had a greater impact than other approaches.

In today's environment of fierce competition, businesses must make numerous modifications. These changes emphasize the provision of higher-quality products and services, as well as the development of new strategies and innovation. Firms that place a stronger emphasis on innovation are more likely to use knowledge management successfully and seek out new skills while sharpening old ones. The complexity and dynamism of the firms' environment can hinder their shift from short-term to long-term success. Businesses can both carry their existing successes into the future and respond to probable future environmental changes due to their ambidexterity.

Chapter 7 highlighted that Software development is one of the most knowledge-intensive jobs possible. Moreover, it requires you to have different kinds of constantly updated information about the software processes themselves, in addition to the products and services you are working on. Software developers repeatedly create various processes for development, which causes software development to be inherently experimental; software engineers thus continually gain knowledge with every development project. Therefore, knowledge management is vital for the software industry.

Knowledge Management covers 4 phases which are: Create, Structure, Share and Apply. Successfully management of these phases leads to successful knowledge management outcomes. Knowledge management has become more efficient by using emerging technologies. These emerging technologies allow the above-mentioned phases of knowledge management to be implemented more effectively.

The Internet of Things (IoT) applications have radically changed our lives by adding great value to the lives of both individuals and organizations. Today, billions of everyday objects are equipped with advanced sensors, wireless networks and innovative computing capabilities. This means that very large data can be transmitted quickly. One of the biggest operational challenges for knowledge management systems is to access the real-time data necessary for optimal and effective decision making. From the moment it emerged, the Internet of Things has made a positive difference in providing high-volume and instant data communication, especially between computer systems, which are one of the basic components of knowledge management systems.

Today, the opportunity to have big data has also led to the need to use advanced technologies in transforming this data into information and knowledge. As the amount of information created and shared increases, the difficulty of discovering information increases in coordination. Artificial intelligence uses modern technologies to simplify the discovery of knowledge. Artificial intelligence powered knowledge bases use new technologies such as semantic search, natural language processing, and machine learning to make it easy for employees to find the information they are looking for quickly and easily. Artificial intelligence powered tools help us consolidate information across multiple systems, making information accessible to all employees, wherever they are.

Artificial intelligence connects data from different sources. Artificial intelligence helps us keep our knowledge base content up to date. Artificial intelligence tools provide key knowledge management metrics. Artificial Intelligence contribute to knowledge management in software industry in some major activities:

- Knowledge distribution: Online databases can provide AIs with knowledge spanning different fields and application areas according to software.
- A well-built machine can extract from the actual data store, which increases with the number of interactions with users feeding new information into the algorithm. This means new information retrieval and therefore a larger data repository for customers or system users.
- The act of delivering (or transferring) knowledge is often performed by chatbots: artificial technologies based on NLP that analyze and interact with human language through a speech-like simulation environment during software development.
- The information caught from the software running on production can automatically be analyzed and fix or improvement areas can be automatically determined.

Chapter 8 highlighted that Software Startups bring innovative products to the market. However, such innovation is at the cost of highly educated guess work about customer expectations and quick decision making by persons responsible for strategic planning and implementation. It is therefore of interest to understand the challenges and practices faced by startups that aim to release something innovative in selected market segments. Hence this paper investigates the challenges faced by entrepreneurs of startups and the practices they follow to become successful. The specific challenges explored include: (i) How startups handle software evolution (ii) Challenges faced in releasing products to the market, and (iii) the state of affairs of Software Engineering in startups. Results indicate that despite guidance and support in terms of well-known and documented development methods, practitioners find it difficult to implement and apply these in practice. They must quickly evolve their products to sustain in the market and the market is highly uncertain which makes the complete process highly probabilistic.

Chapter 9 pertains to the innovations in requirement prioritisation process. Software requirements prioritisation is an important task that ultimately determines whether the software is successful and achieves customer satisfaction. Most software projects have a large number of requirements, so there is a need to prioritise which requirements to include. Startups use agile methodologies to deliver innovative software solutions, as agile adapts to requirement changes well and delivers software quickly in short increments, called sprints. Benefits may be more notable for smaller companies and startups as they tend to have a greater focus on the customer and on process improvement, whereas large companies may suffer from a rigid organisation and functional silos. The product owner is responsible for managing and prioritising a dynamic product backlog, to reflect the continuous re-prioritisation of the requirements. Developers are often delegated this decision-making role, particularly for small organisations or startups who may not have IT domain knowledge and cannot afford an IT consultant to act on their behalf. However, there is little research about the practices of agile requirements re-prioritisation, the activity to reprioritise requirements at the start of each sprint. This research contributes to this gap by identifying the factors considered when prioritising requirements for five popular prioritisation approaches. This research also compares these factors to the agile requirements re-prioritisation process to see how well these popular approaches support the agile process. The five popular requirements prioritisation approaches are Analytic Hierarchy Process (AHP), Quality Functional Deployment (QFD), planning game, binary search tree, and $100 allocation. Framework synthesis was used to identify a best-fit framework developed by a robust methodology, and relevant for the agile requirements re-prioritisation process. The chosen best-fit framework considers six factors when prioritising requirements, business value, risk, effort estimation, learning experience, external change, and project constraints. First. the factors considered by the five popular approaches were identified. The results show that five factors were reported in literature for the planning game, three were reported for AHP, one was reported for QFD and no factors were identified for binary search tree and $100 allocation. Although, the factor business value was not identified in the literature for $100 allocation or binary search tree, it is likely that stakeholders consider business value for $100 allocation as they allocate more dollars to the requirements, which are more important. It is also likely that business value is considered for binary search tree, while determining the placement of each candidate requirement on the tree. Second, the factors from the agile requirements re-prioritisation process were compared with the factors considered by the five popular approaches. The results confirm five of the factors identified in the agile requirements re-prioritisation process, the sixth factor external change, was not reported in the literature for the five popular approaches. The planning game covers five of the factors whereas AHP covers three of the factors. QFD only covered one factor and both the binary search tree and £100 allocation approaches did not report any of the factors. Although, the Binary search tree and $100 allocation approaches have numerous benefits, including being fast and easy to use. This may influence the choice of approach used for agile requirements re-prioritisation. This study contributes insights that are important for requirements prioritisation literature and practice.

Chapter 10 pertains to innovation in risk management. Risk is an inherent part of a startup journey, and software startups need to deal with different type of risks, including technical and product risks. In established companies, risk management is well-established research and practice area, and proof to be helpful for successfully managing software development projects. However, it is less known in a software startup context whether risk management also work as they are in established contexts. This paper reports a result from qualitative studies in nine software startups in Denmark and Finland. The outcomes indicates that startups founders do not believe in risk management methods and prioritize other tasks on their to-do list. These findings might not be generalized for a larger population; however, they could be useful for startups companies in Nordic countries, which share similar environmental contexts with our cases. We believe that the insights from this study would be helpful for people who are doing or want to start their software business. However, there is a need to further explore if there is any impact on startup performance when risk management is used versus when it is not.

Chapter 11 highlights the importance of scholarly literature on startup capacities to stimulate innovation in pandemic times is highlighted in this chapter. The scholarly literature can help startups looking for opportunities or solutions in the face of a pandemic, but knowledge acquisition from secondary materials may be limited due to the growing number of publications, retractions, and Preprints. The growing number of publications and venues makes it more difficult for entrepreneurs to get the information they need, analyse it, and then use collective intelligence to turn it into useful business knowledge. Retractions may steer startups in the incorrect direction, resulting in a waste of financial resources. Preprints are non-peer reviewed research articles that may provide some direction to startups but should not be relied upon entirely. The solutions to these issues are finally provided. Addressing these concerns could make scholarly literature beneficial to startups, allowing the global community to respond to the pandemic as a whole.

Chapter 12 reported the innovative IT-technologies in the field of mechanical engineering, allowing to increase the efficiency of production. This chapter reflects a particular task of automation of a particular branch of mechanical engineering - the technology of mechanical engineering. New methods of calculation of typical multivariable tasks are considered, as well as the effectiveness of the introduction of automation at the level of the design office.

Chapter 13 pertains to process innovation in requirement prioritisation. Agile software development is popular among startup companies, who quickly develop software with a focus on innovation. Software can be developed for a variety of applications, including mobile phones and the controls of an aeroplane. Prioritisation is an essential process of any software development project, as there are usually more requirements than there is time and budget. There are various approaches available, to help decide which requirements to prioritise for inclusion in the software. The wrong approach could waste resources and cause customer dissatisfaction. There are also constraints for startups, such as small teams and multiple influencers which must be considered when choosing a suitable approach. An awareness of limitations with prioritisation approaches could help inform software developers with this decision. However, there is limited research linked to the limitations of requirements prioritisation approaches. This research helps to address this gap by identifying limitations for five popular approaches. The five requirements prioritisation approaches studied were Analytic hierarchy process (AHP), quality functional deployment (QFD), the planning game, binary search tree, and $100 allocation. A search of academic literature was conducted to identify sentences and paragraphs describing the limitations. With little research on prioritisation approach limitations, Grounded Theory was chosen. Verbatim text about the limitations was inductively analysed to identify which were reported for each of the five popular prioritisation approaches. The findings contributed sixteen limitations associated with the five popular prioritisation approaches. Nine limitations for AHP and QFD, seven for the planning game, six for $100 allocation, and four for binary search tree. While analysing these limitations dependencies were reported among them. For example, the quality of the requirements limitation could impact the validity issues limitation. Therefore, this study also contributes a framework showing these dependencies, how the limitations can impact or influence other limitations. The results could help software developers to understand the limitations of each approach and inform the approach they choose for requirements prioritisation. With the fewest limitations, this study shows that the binary search tree could be the best approach. However, an approach with a high number of limitations may be preferred if the benefits outweigh the limitations. Therefore, further research is needed to provide a balanced view, and also consider the benefits of these five popular approaches. Future research could also be used to verify the framework.

Chapter 14 pertains to technology enhanced business model innovation. Innovation is critical for any forward-thinking organization. This is where technology plays a major role. Choosing technologies that will empower an organization is challenging. Even a good development strategy needs to be implemented properly. To innovate enough, start thinking about what kind of technology is actually required in order to be benefited with outcomes. Information technology (IT) innovation in an enterprise involves using technology in new ways to create a more efficient organization and improve alignment between technology initiatives and business goals. IT innovation can take many forms like turning business processes into automated IT functions, developing applications that open new markets, or implementing desktop virtualization to increase manageability and cut hardware costs. Information and Communication Technologies (ICT) are emerging as a promising paradigm for creating a profound change in digitizing technologies. Technology innovation can take many forms, for instance, novel software implementing new algorithms and data processing models; or new hardware components (sensors, processors, components); or improved user interfaces offering seamless experiences; it can also happen at a higher level, in the form of new processes, business models, monetization engines, and so on.

To bring in technology into business model entrepreneurs must involve themselves into research and development (R&D), generating new ideas, conducting experiments, designing and implementing new changes into the system. To achieve better performance appropriate strategy has to be followed. To bring in technology into business the first step of the entrepreneur must be recognizing the unanswered or unresolved customer needs. There are three characteristics to be considered for technology with respect to the business model development. Technology supports business model through various supporting functions for a specific business model. Technology acts as the enabler for a business model and business model enables an innovative technology.

Both innovation and technology are tightly interlaced. Two very notable ways technology propels innovation forward is that it boosts tinkering and experimentation, and that in itself accelerates innovation processes. Earlier experimentation with new technologies was only possible by multinational corporations or government-funded research labs. Today, affordable technology digital and others make it possible for most enterprises big and small to experiment with ideas and concepts in whole new ways, and also in reality instead of only in test labs. Innovation must be socially desirable, economically profitable, and technologically feasible.

Technology, Innovation and Ventures capabilities should be brought together to support the clients' needs for sustainable growth. Approaches to anticipate the new trends, assess their potential, validate their enterprise-readiness, and exploit them responsibly should be enabled. Applied innovation in industries has enabled scaling, with certainty and trust, and with the power of data and intelligence built in.

This book includes research articles on several forms of startup innovations, such as process innovations, business model innovations, and product innovations. The expertise imparted by the book will assist its readers in adapting the knowledge to their startup context in order to overcome difficulties that are specific to their situation. Furthermore, the book makes a significant contribution to the body of knowledge by expanding on innovation-specific knowledge in the context of startups. This book will be especially valuable to startups with a high failure rate and minimal resources. By embracing an interdisciplinary approach integrating computer engineering and business management, this book provides a good range of research studies to stimulate further research in innovation management relevant to startups. Interdisciplinary solutions are needed by the startup community to be inventive and gain a lasting competitive edge in extremely dynamic markets. This book will be an amazing source of instant knowledge for startups, boosting their innovative capabilities and success rates, with a perfect blend of empirical research and evaluation study kinds.

The editors hope that the intended audience will benefit from this book, and we wish them a Happy Reading, Learning, and Adoption.

*Varun Gupta*
*Universidad de Alcalá, Madrid, Spain*

*Chetna Gupta*
*Jaypee Institute of Information Technology, Noida, India*

# Acknowledgment

The editors would like to express their gratitude to everyone who contributed to this project, especially the authors and reviewers who participated in the review process. This book would not have been possible without their help.

First and foremost, the editors would like to express their gratitude to all of the contributors for their contributions. The authors of the chapters who offered their time and expertise to this book have our heartfelt gratitude.

Second, the editors like to thank the reviewers for their substantial contributions to improving the quality, coherence, and content presentation of the chapters. The majority of the authors also worked as referees, which we greatly appreciate.

# Chapter 9: Use of framework synthesis to identify the factors considered for five popular prioritisation approaches.

Zoe Hoy[1]

1 Portsmouth Business School, University of Portsmouth, Portsmouth PO1 3DE, UK

Abstract

Software requirements prioritisation is an important task that ultimately determines whether the software is successful and achieves customer satisfaction. Startups use agile methodologies to develop software, as it adapts to requirement changes well and delivers software quickly in short increments, called sprints. However, there is little research about the practices of agile requirements re-prioritisation, the activity to reprioritise requirements at the start of each sprint. This research contributes to this gap by identifying the factors considered for five popular prioritisation approaches and compares them to the agile requirements re-prioritisation process. Framework synthesis is used to compare a best-fit framework with the five popular approaches. The results show that the approaches studied do not address all factors of the agile requirements re-prioritisation process. The planning game covers five of the factors whereas Analytical Hierarchy Process covers three of the factors. This may influence the choice of approach used for agile requirements re-prioritisation. This study contributes insights that are important for requirements prioritisation literature and practice.

## 1 Introduction

Startups use agile software development (Gupta, Fernandez-Crehuet, Hanne, & Telesko, 2020; Lim, Bentley, & Ishikawa, 2020; Nurdiani, Jabangwe, & Petersen, 2016) to deliver innovative software solutions (Gupta et. al., 2020). Agile engages stakeholders, supports requirements changes and delivers software quickly (Luong, Sivarajah, & Weerakkody, 2021).

Requirements prioritisation is an important activity for a software project. Requirements need to be prioritised which are high value and innovate the software (Gupta et al., 2020), but the activity is also challenging. The right balance must be achieved among competing requirements to ensure that the software meets the customer's needs (Svensson et al., 2011). Most software projects have a large number of requirements, so there is a need to prioritise which requirements to include in each sprint due to limited resources such as time and money (Hudaib, Masadeh, Qasem & Alzaqebah, 2018).

For agile, the requirements are held in a product backlog. The product backlog is dynamic as it reflects the continuous re-prioritisation of the requirements. Uncertainties in the form of requirement changes, are addressed with the focus on business value, incremental deliveries of the software and continuous re-prioritisation of the requirements (Racheva, Daneva, Herrmann, & Wieringa, 2010a). For agile software development, the product owner is responsible for managing and prioritising the product backlog (Bass, 2013). Racheva, Daneva, Sikkel, Wieringa. and Herrmann (2010b) found that developers are often delegated this client decision-making role, particularly for small client organisations or startups who may not have IT domain knowledge and cannot afford an IT consultant to act on their behalf.

Prioritising requirements is an ongoing activity for agile software development, as re-prioritisation of the requirements (Racheva et al., 2010a) occurs at the start of each sprint, to reflect the changing client's needs. Sprints are short software development cycles, typically 2-4 weeks in duration, where the customer/product owner is a member of the team (van Waardenburg & van Vliet, 2013).

There are many approaches which can be used to prioritise requirements. Achimugu, Selamat, Ibrahim and Mahrin (2014) identified 49 approaches. The article has been well cited and reports the top five most cited and used approaches as Analytic Hierarchy Process (AHP), Quality Functional Deployment (QFD), planning game, binary search tree, and $100 allocation. These five popular approaches have been selected for this study.

AHP is a robust, rigorous and proven method to evaluate alternatives (Das & Mukherjee, 2008; de Felice & Petrillo, 2010). Complex decisions are presented in a hierarchical structure, with the goal at the first level, criteria in the second level and sub-criteria in the third level (Acharya, Sharma & Gupta, 2018). This reduces complex decisions to a number of pair-wise comparisons, to provide the best alternative (de Felice & Petrillo, 2010). The large number of pairwise comparisons makes the process unsusceptible to judgement error (Karlsson, Wohlin, & Regnell, 1998). A ratio scale is used to rank the alternatives by their relative weights (de Felice & Petrillo, 2010; Dabbagh, Lee & Parizi, 2016), which provides a useful assessment of the requirements (Karlsson et al., 1998). The AHP approach can be used in combination with the QFD approach (Das & Mukherjee, 2008; de Felice & Petrillo, 2010; Akao, 2014). Akao (2014) used AHP to calculate the degree of importance weightings for each of the demanded quality items. The degree of importance weightings, reflects the importance of the requirements.

QFD is an approach to assure that the customers' needs are incorporated into a new software product (Akao & Mazur, 2003). The software product specification (hows) are based on the customers' needs (whats) and analysis of competitors (whys) (de Felice & Petrillo, 2010). QFD has been used around the world since 1966 (Akao & Mazur, 2003). An importance rating is calculated for each requirement. Those with a higher importance rating are prioritised as more important.

The planning game is easy and quick to use, and scalable for a large number of requirements (Ahl, 2005). The business value of the requirements and project constraints are considered (Maurer & Martel, 2002) when prioritising requirements, providing a balance between the customers' needs and the expertise and experience of the development team members (Maurer & Martel, 2002). Karlsson, Thelin, Regnell, Berander and Wohlin (2007) found the planning game quicker to prioritise requirements and more accurate, when compared to pairwise approaches like AHP.

Ahl (2005) conducted an experiment to compare five prioritisation approaches; $100 test, AHP, binary search tree, planning game and planning game with AHP. The results showed that binary search tree was the best prioritisation approach having the benefits of scalability; to handle a medium/large number of requirements, ease of use and accuracy. These benefits were also reported by Saghir and Mustafa (2016) and Hudaib et al. (2018).

For the $100 allocation approach, stakeholders allocate $100 to candidate requirements (Chatzipetrou, Rovegård., & Wohlin, 2010). Requirement priorities are on a ratio scale, showing the relative importance between the requirements (Solinski & Petersen, 2016). The ratio scale enables the stakeholders' dollar allocations to be evaluated, to detect whether there is disagreement between the stakeholders or whether there are clusters of stakeholders with similar views (Riņķevičs & Torkar, 2013). When there is disagreement between the stakeholders, a knowledge of specific needs for a group of stakeholders, can help the product owner to make an informed decision when they prioritise the product backlog (Sverrisdottir, Ingason, & Jonasson, 2014). The approach is fast, easy to use, perceived to provide accurate results (Ahl, 2005; Hudaib, et al., 2018) and is scalable for a medium number of requirements (Hudaib et al., 2018).

Cristiano, Liker and White (2001) claim that benefits are more notable for smaller companies. Ettlie and Johnson (1994) claim that smaller companies have a greater focus on the customer and on process improvement, whereas large companies may suffer from a rigid organisation and functional silos. This means that the benefits of the five popular approaches may be more remarkable for smaller companies and startups.

Racheva et al. (2010a) claim that very little is known about the practices of agile re-prioritisation. They have contributed towards this gap by developing a conceptual model for agile requirements re-prioritisation. However, a limitation acknowledged by the authors is that it is a first proposal for the conceptual model, developed from literature (Racheva et al., 2010a) and therefore cannot explain how requirements prioritisation decision-making takes place. This study will contribute further in this area by identifying the factors considered when prioritising requirements for five popular prioritisation approaches. This research also compares these factors to the agile requirements re-prioritisation process to see how well these popular approaches support the agile process.

The need to increase knowledge of these agile re-prioritisation practices has motivated the following research questions (RQ):

RQ1: What factors do the five popular prioritisation approaches consider when prioritising requirements?

RQ2: How well do the five popular prioritisation approaches support the agile requirements re-prioritisation process?

The structure of this paper is as follows. First a best-fit framework is chosen for the agile requirements re-prioritisation process. Then factors considered by the five popular prioritisation approaches are identified and compared with the requirements re-prioritisation process. Before there are conclusions and a reflection on the contribution to knowledge.

## 2 Choosing a best fit framework

To choose the agile re-prioritisation process for this study various frameworks are evaluated to determine the best fit framework. The methodology used to develop each framework is critically evaluated, to choose a framework developed by a robust methodology, and relevant for the agile requirements re-prioritisation process. The factors of the chosen agile re-prioritisation process are introduced in section 2.2 and used as themes for the data analysis.

### 2.1 Choosing a framework for agile requirements re-prioritisation

Various frameworks have been reported in literature for requirements prioritisation. In the conceptual model developed by Al-Ta'ani and Razali (2013), two criteria are identified for requirements prioritisation; project constraints, which includes schedule, budget and scope, and requirements nature, which includes visibility, business value, dependencies and complexity. However, there are two limitations for this study, firstly more details were needed about the research methodology. For example, the search terms used, and criteria for the included studies and the steps followed to conduct the content analysis. These details are needed so that a researcher can replicate the study and achieve the same results (Saunders, Lewis, & Thornhill, 2016). Secondly, in order to ensure the validity of the study, the articles included must be fit for purpose (Denyer, & Tranfield, 2011). It was not clear how the quality of the included articles was assured as the details of the method for this were not evident in the reported study.

Al-Ta'ani and Razali (2016) developed another framework in a later study, which included the same two criteria for requirements prioritisation as their earlier study, i.e. project constraints and requirements nature. This framework was specifically developed for agile requirements prioritisation. However, this study had limited details of the methodology followed. For example, it was stated that grounded theory was used (Al-Ta'ani & Razali, 2016), but very limited details of the method were provided to enable the study to be replicated (Saunders et al., 2016).

A study by Moisiadis (2002) identified a framework for prioritising requirements, but not for agile requirements prioritisation. Agile requirements prioritisation is very different to plan-driven development (Racheva et al., 2010a). Agile has a product owner role and requirements are re-prioritised for every iteration of the software development.

The Conceptual Model B developed by Racheva et al. (2010a) was chosen as the best-fit framework for this research as it was specifically developed for agile requirements prioritisation. It was based on the description of 22 requirements prioritisation approaches listed in Racheva et al. (2010a)'s Table 1, with a clearly described method that used quality criteria to ensure the quality of literature sources included in the study. A limitation of the conceptual model, acknowledged by the authors, is that it has not been empirically validated (Racheva et al., 2010a). However, this limitation was not of a concern for this study, which compared the model to literature sources written by the creators of five popular prioritisation approaches. Any new factors not addressed by the best-fit framework would be analysed using thematic analysis.

## 2.2 Factors of the chosen framework

The best-fit framework considers six factors when prioritising requirements; business value, risk, effort estimation, learning experience, external change, and project constraints (Racheva et al., 2010a).

The business value for each story is determined by the client or product owner (Achimugu et al., 2014) and used to prioritise the requirements in the product backlog. The business value assigned reflects the product owner's tacit knowledge about the requirement and their learning experience, especially during the re-prioritisation of requirements (Racheva et al., 2010b). Racheva et al. (2010b) also identified that negative value can be used instead of business value. Negative value considers the potential lost business value of not implementing the requirement, instead of the business value to be gained by implementing the requirement.

The development team determines the risk (Achimugu et al., 2014) and effort for each requirement and provides this information to the product owner/clients for their decision making (Racheva et al., 2010a). In the planning game, requirements are assigned one of three risk categories, category one is for requirements which the development team are very confident they can estimate precisely, category two is for requirements which they are confident they can estimate reasonably well, and category three is for requirements which they cannot estimate (Beck, 2000). This risk reflects the perceived level of uncertainty with the estimation for each individual user story (Racheva et al., 2010a). Requirements are prioritised for the next release based on their business value, risk and the team's velocity (Beck, 2000).

The size or effort to implement requirement can be expressed in story points and estimated using planning poker cards (Grenning, 2002). The requirement is described to the agile team and questions are asked by the team members to gain further clarification as needed. Each team member has a deck of planning poker cards and holds up one card to the other team members, which they perceive represents the effort to implement the requirement. The team members then compare cards and discuss the reasons for their choice. Further estimation rounds continue until consensus is reached for the effort needed.

The factor learning experience is where the client uses their knowledge and experience in the agile software project to assess the business value for requirements (Racheva et al., 2010a).

The factor external changes could impact requirements prioritisation, for example Racheva et al. (2010a) claim that external changes from the project's or company's context, could influence which requirements are prioritised.

The output from requirements prioritisation is a prioritised product backlog, which is an input for the sprint (iteration) planning meeting. The sprint planning meeting is held at the start of each software development cycle (Racheva et al., 2010a. The purpose of the meeting is to prioritise which requirements from the prioritised product backlog should be included in the next sprint. Constraints such as the velocity, the available capacity which the agile team have for a single sprint are considered (Rosenberger & Tick, 2021). Requirements are chosen for the next sprint and moved into the sprint backlog, ensuring that the velocity is not exceeded.

## 3 Research Method

The framework synthesis approach was chosen as a published conceptual model was available to compare with the five popular approaches. Framework synthesis is a highly structured method to synthesize qualitative data (Carroll, Booth, & Cooper, 2011). A distinct factor of framework synthesis is the framework of best-fit, which is used (Barnett-Page & Thomas, 2009). Themes are identified from the framework and used as codes (Carroll et al., 2011) to guide the data extraction and synthesis of the findings (Shaw, Holland, Pattison., & Cooke, 2016).

Framework synthesis is a pragmatic method (Carroll et al., 2011) following both a deductive and inductive approach. A deductive approach is followed as it builds on the existing framework (Carroll et al., 2011). Then an inductive approach is followed to analyse new topics, which may emerge from the data (Barnett-Page & Thomas, 2009; Carroll et al., 2011) that cannot be mapped to the themes is performed. In framework synthesis, thematic analysis is used to analyse any new topics that may emerge (Carroll, Booth, Leaviss, & Rick, 2013). This is a popular approach for synthesis in software engineering (Cruzes & Dybå, 2011a) and provides a structured approach to develop themes from recurring patterns (Cruzes & Dybå, 2011b).

One limitation of framework synthesis is its reliance on a suitable best fit framework being identified (Carroll et al., 2011). As recommended by Carroll et al. (2013) the [1]BeHEMoTH strategy was used to find a suitable framework. Search terms included prioritization process and prioritisation process to represent the behaviour of interest and context (BeH) and the term framework was used to represent Models or Theories (MoTH). For exclusions (E), studies were manually excluded from the research results, which did not include a model for the requirements prioritisation process.

The model developed by Racheva et al. (2010a) was the best-fit framework chosen for this research in section 2.1, as it was developed by a robust methodology and focused on agile software development. The factors for the model described in section 2.2 were the themes: business value, risk, effort estimation, learning experience, external change, and project constraints (Racheva et al., 2010a).

A second limitation of framework synthesis is whether the author correctly interpreted themes consistently with their original intended meaning (Carroll et al., 2011). However, as Racheva et al. (2010a) provided a detailed description for each factor in the model, this limitation is minimal.

The literature sources selected were those written by the creators of five popular prioritisation approaches; Leffingwell and Widrig (2000) for $100 allocation, Saaty (2003; 2008) for AHP, Hibbard (1962) for binary search tree, Beck (2000) for planning game and Akao (2014) and Kamisawa (1994) edited by Akao for QFD. These original sources were chosen to prevent bias from any variations published on the approaches. These texts were analysed to identify sentences or paragraphs describing the prioritisation process, which were then compared to the themes. The five steps for thematic synthesis (Cruzes & Dybå, 2011b) were used to analyse any new factors, that emerged from the data (Barnett-Page & Thomas, 2009; Carroll et al., 2011).

## 4 Results and discussions

In this section, the factors considered by five popular prioritisation approaches are presented and then compared to the factors of the agile requirements re-prioritisation process.

[1] "BeHEMoTH (Be – Behaviour of interest, H – Health context, E – exclusions, MoTH – Models or Theories)" Carroll et al. (2013).

For RQ1, Table *1* identifies the factors considered by the five popular prioritisation approaches. Five factors were reported in literature for the planning game, three were reported for AHP, one was reported for QFD and no factors were identified for binary search tree and $100 allocation.

| Factors identified in literature for the approaches (✓) | AHP | Binary Search Tree | Planning Game | QFD | $100 allocation |
|---|---|---|---|---|---|
| Business Value | ✓ | | ✓ | ✓ | |
| Risk | ✓ | | ✓ | | |
| Effort | ✓ | | ✓ | | |
| Learning experience | | | ✓ | | |
| Project constraints | | | ✓ | | |

Table 1: Comparison of the approaches against the factors of the re-prioritisation process

For RQ2 the factors considered by the five approaches were compared to the factors considered by the agile requirements re-prioritisation process. Literature for the five popular approaches did not consider the factor external change, hence this factor was excluded from Table *1*. Also, no new factors emerged from the analysis of the five popular approaches, therefore no new factors were added to Table *1*.

For RQ2, Table 1 shows that the five approaches studied do not address all factors of the agile requirements re-prioritisation process. The planning game covers five of the factors of the re-prioritisation process. AHP covers three of the factors. Although AHP literature (Saaty, 2003; 2008) did not report on project constraints, project constraints could be chosen as a criterion. For AHP, stakeholders choose the criteria to prioritise the requirements against (Saaty, 2008).

The factor business value was not identified in the literature for $100 allocation or binary search tree. However, it is likely that stakeholders consider business value for '$100 allocation' as they allocate more dollars to the requirements, which are more important (Leffingwell & Widrig, 2000). Requirements of higher importance could be considered to have higher business value. It is also likely that business value is considered for binary search tree, while determining the placement of each candidate requirement on the binary search tree. For example, those requirements on the right-hand side of the tree are more important than those requirements on the left-hand side of the tree (Hibbard, 1962). The right side of the tree could be considered to have a higher business value.

## 5 Conclusion

For RQ1, Table 1 provides an understanding of factors, which the five popular approaches consider. Five factors were reported in literature for the planning game, three were reported for AHP, one was reported for QFD and no factors were identified for binary search tree and $100 allocation. Although, the factor business value was not identified in the literature for $100 allocation or binary search tree, it is likely that stakeholders consider business value for $100 allocation as they allocate more dollars to the requirements, which are more important. It is also likely that business value is considered for binary search tree, while determining the placement of each candidate requirement on the tree.

For RQ2, the factors from the agile requirements re-prioritisation process were compared with the factors considered by the five popular approaches. The results confirm five of the factors identified in the agile requirements re-prioritisation process, the sixth factor external change, was not reported in the literature for the five popular approaches. The planning game covers five of the factors whereas AHP covers three of the factors. QFD only covered one factor and both the binary search tree and £100 allocation approaches did not report any of the factors. This may influence the choice of approach used for agile requirements re-prioritisation.

Table 1 shows the planning game fits the requirements re-prioritisation process the best. The planning game (Beck, 2000) identified five factors of the agile re-prioritisation process, except external change. The suitability of the planning game approach for agile software development is supported by Wood, Michaelides and Thomson (2013) who claim that the activity customer planning, which includes the planning game is positively related to the performance of agile software development teams.

Although, the Binary search tree and $100 allocation approaches did not support any of the factors listed in Table 1, they have numerous benefits, including being fast and easy to use. These benefits could support startups and smaller companies, which are more adaptive and have a greater focus on the customer and process improvement.

As little is known about the practices of agile re-prioritisation our study has contributed towards this gap, with insights that are important for requirements prioritisation literature and practice. Further research is needed to explore these five popular approaches in startups using agile software development.

## References


Acharya, V., Sharma, S. K., & Gupta, S. K. (2018). Analyzing the factors in industrial automation using analytic hierarchy process. *Computers and Electrical Engineering*, http://doi.org/10.1016/j.compeleceng.2017.08.015

Achimugu, P, Selamat, A., Ibrahim, R., & Mahrin, M. N. (2014). A systematic literature review of software requirements prioritization research. *Information and software technology*, 56(6), 568-585. http://doi.org/10.1016/j.infsof.2014.02.001

Ahl, V. (2005). *An experimental comparison of five prioritization methods – Investigating ease of use, accuracy and scalability* (Unpublished master's dissertation). http://www.diva-portal.org/smash/get/diva2:833611/FULLTEXT01.pdf

Akao, Y. (2014). The method for motivation by Quality Function Deployment (QFD). *Nang Yan Business Journal*, *1*(1), 1-9. http://doi.org/10.2478/nybj-2014-0001

Akao, Y., & Mazur, G. H. (2003). "The leading edge in QFD: past, present and future". *International Journal of Quality & Reliability Management*, *20*(1), 20-35. http://doi.org/10.1108/02656710310453791

Al-Ta'ani, R. H., & Razali, R. (2016). A Framework for Requirements Prioritisation Process in an Agile Software Development Environment: Empirical Study. *International Journal on Advanced Science, Engineering and Information Technology*, *6*(6), 846-856. http://doi.org/10.18517/ijaseit.6.6.1375

Al-Ta'ani, R. H., & Razali, R. (2013). Prioritizing requirements in agile development: A conceptual framework. *Procedia Technology, 11*(2013), 733-739. http://doi.org/10.1016/j.protcy.2013.12.252

Barnett-Page, E., & Thomas, J. (2009). Methods for the synthesis of qualitative research: a critical review. *BMC Medical Research Methodology*, 9(1), 59-69. http://doi.org/10.1186/1471-2288-9-59

Bass, J. M. (2013). Agile Method Tailoring in distributed Enterprises: Product Owner Teams. *Proceedings of IEEE 8th International conference on Global Software Engineering (ICGSE)*, 154-163. http://doi.org/10.1109/ICGSE.2013.27

Beck, K. (2000). *Extreme Programming Explained*. Addison-Wesley.

Carroll, C., Booth, A., Leaviss, J., & Rick, J. (2013). "Best fit" framework synthesis: refining the method. *BMS Medical Journal Methodology*, 13(37), 1-16. http://doi.org/10.1186/1471-2288-13-37

Carroll, C., Booth, A., & Cooper, K. (2011). A worked example of "best fit" framework synthesis: A systematic review of views concerning the taking of some potential chemopreventive agents. *BMS Medical Journal Methodology*, 11(29), 1-9. http://doi.org/10.1186/1471-2288-11-29

Chatzipetrou, P., Rovegård, P., & Wohlin, C. (2010). Prioritization of issues and requirements by Cumulative Voting: A compositional data analysis framework. *36th EUROMICRO Conference on Software Engineering and Advanced Applications*. 361-370. http://doi.org/10.1109/SEA.A.2010.35

Cristiano, J. J., Liker, J. K., & White, C. C. (2001). Key factors in the successful application of Quality Function Deployment (QFD). *IEEE Transactions on Engineering Management*, 48(1). 81-95. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=913168

Cruzes, D. S., & Dybå, T. (2011a). Research synthesis in software engineering: A tertiary study. *Information Systems*, 53(5), 440-455. http://doi.org/10.1016/j.infsof.2011.01.004

Cruzes, D. S., & Dybå, T. (2011b). Recommended Steps for Thematic Synthesis in Software Engineering. *2011 International Symposium on Empirical Software Engineering and Measurement*, 275-284. http://doi.org/10.1109/ESEM.2011.36

Dabbagh, M., Lee, S. P., & Parizi, R. M. (2016). Functional and non-functional requirements prioritization: empirical evaluation of IPA, AHP-based, and HAM-based approaches. *Soft Computing*, *20*(11), 4497-4520. http://doi.org/10.1007/s00500-015-1760-z

Das, D., & Mukherjee, K. (2008). Development of an AHP-QFD framework for designing a tourism product. *International Journal of Services and Operations Management*, 4(3), 321-344. http://doi.org/10.1504/IJSOM.2008.017297

Denyer, D., & Tranfield, D. (2011). Producing a Systematic Review. In D. A. Buchanan, & A. Bryman (Eds.), *The SAGE Handbook of Organizational Research Methods* (pp. 671-689). Sage.

Ettlie, J. E. (1994). Product development benchmarking versus customer focus in applications of Quality Function Deployment. *Marketing Letters*, 5(2). 107-116. http://search.ebscohost.com/login.aspx?direct=true&db=edsjsr&AN=edsjsr.40216330&site=eds-live

de Felice, F., & Petrillo, A. (2010). A multiple choice decision analysis: an integrated QFD – AHP model for the assessment of customer needs. *International Journal of Engineering, Science & Technology*, *2*(9), 25-38. https://www.ajol.info/index.php/ijest/article/view/63849

Grenning, J. W. (2002). *Planning Poker or how to avoid analysis paralysis when release planning*. https://wingman-sw.com/papers/PlanningPoker-v1.1.pdf

Gupta, V., Fernandez-Crehuet, J. M., Hanne, T., & Telesko, R. (2020). Requirements engineering in software startups: a systematic mapping study. *Applied Sciences*, 10(6125), 1-19. http://doi.org/10.3390/app10176125.

Hibbard, T. N. (1962). Some combinatorial properties of certain trees with applications to searching and sorting. *Journal of the ACM*, 9(1), 13-28. http://doi.org/10.1145/321105.321108

Hudaib, A., Masadeh, R., Qasem, M. H., & Alzaqebah, A. (2018). Requirements Prioritization Techniques Comparison. *Modern Applied Science*, 12(2), 62-80. http://dx.doi.org/10.5539/mas/v12n2p62

Karlsson, L., Thelin, T., Regnell, B., Berander, P., & Wohlin, C. (2007). Pair-wise comparisons versus planning game partitioning – experiments on requirements prioritisation techniques. *Empirical Software Engineering*, 12(1), 3-33. http://doi.org/10.1007/s10664-006-7240-4

Karlsson, J., Wohlin, C., & Regnell, B. (1998). An evaluation of methods for prioritizing software requirements. *Information and Software Technology*, 39(14-15), 939-947. http://ac.els-cdn.com/S0950584997000530/1-s2.0-S0950584997000530-main.pdf?_tid=d69aba2c-9187-11e7-8066-00000aab0f27&acdnat=1504540006_4087f654bed319e8e0d6cf66ef452870

Kamisawa, N. (1994). The use of prioritization in quality deployment at the planning and design stages. In S. Mizuno, & Y. Akao (Eds.), *QFD The customer-driven approach to quality planning and deployment* (pp. 108-134). Asian Productivity Organization.

Leffingwell, D., & Widrig, D. (2000). *Managing Software Requirements: A unified approach*. Addison-Wesley.

Luong, T. T., Sivarajah, U., & Weerakkody, V. (2021). Do agile managed information systems projects fail due to a lack of emotional intelligence? *Information Systems Frontiers*, 23, 415–433. https://doi.org/10.1007/s10796-019-09962-6

Lim, S. L., Bentley, P. J., & Ishikawa, F. (2020). Reaching the Unreachable A method for early stage software startups to reach inaccessible stakeholders within large corporations. *2020 IEEE 28th International Requirements Engineering Conference*, 376-381. http://doi.org/10.1109/RE48521.2020.00051

Maurer, F., & Martel, S. (2002). Extreme Programming: Rapid development for web-based applications. *IEEE Internet Computing*, 6(1). 86-90. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=989006

Moisiadis, F. (2002). The fundamentals of prioritising requirements. *Systems Engineering, Test & Evaluation Conference*, 1-12. https://pdfs.semanticscholar.org/7395/4e283497f15c6b0d7d2ef615afbd1587450c.pdf

Nurdiani, I., Börstler, J., & Fricker, S. A. (2016). The impacts of agile and lean practices on project constraints: A tertiary study. *The Journal of Systems and Software*, 119(2016), 162-183. http://ddo.org/10.1016/j.jss.2016.06.043

Racheva, Z., Daneva, M. Herrmann, A., & Wieringa, R. (2010a). A conceptual model and process for client-driven agile requirements prioritization. *4th International Conference on Research Challenges in Information Science*, 287-298. http://doi.org/10.1109/RCIS.2010.5507388

Racheva, Z., Daneva, M. Sikkel, K., Wieringa, R., & Herrmann, A. (2010b). Do we know enough about requirements prioritization in agile projects: insights from a case study. *2010 18th IEEE International Requirements Engineering Conference*, 147-156. http://doi.org/10.1109/RE.2010.27

Riņķevičs, K., & Torkar, R. (2013). Equality in cumulative voting: A systematic review with an improvement proposal. *Information and Software Technology*, 55(2), 267-287. http://doi.org/10.1016/j.infsof.2012.08.004

Rosenberger, P., & Tick, H. J. (2021). Agile enhancement of critical PMBOK V6 processes. *Journal of Modern Project Management*, 9(1), 190-203. https://doi.org/10.19255/JMPM02613

Saaty, T. L. (2008). Decision making with the analytic hierarchy process. *International Journal of Services Sciences*, *1*(1), 83-98. https://doi.org/10.1504/IJSSci.2008.01759

Saaty, T. L. (2003). Decision making with the AHP: Why is the principal eigenvector necessary. *European Journal of Operational Research*, *145*(1), 85-91. https://doi.org/10.1016/S0377-2217(02)00227-8

Saghir, S., & Mustafa, T. (2016). Requirements Prioritization Techniques for Global Software Engineering. *Pakistan Journal of Engineering, Technology & Science*, 6(1), 42-63. http://journals.iobmresearch.com/index.php/PJETS/article/view/1143

Saunders, M., Lewis, P., & Thornhill, A. (2016). *Research Methods for Business Students* (7th ed.). http://lib.myilibrary.com/Open.aspx?id=819487

Shaw, R. L., Holland, C., Pattison, H. M., & Cooke, R. (2016). Patients' perceptions and experiences of cardiovascular disease and diabetes prevention programmes: A systematic review and framework synthesis using the Theoretical Domains Framework. *Social Science & Medicine*, 156(2016), 192-203. http://doi.org/10/1016/j.socscimed.2016.03.015

Solinski, A., & Peterson, K. (2014). Prioritizing agile benefits and limitations in relation to practice usage. *Software Quality Journal*, 24(2016), 447-482. http://doi.org/10.1007/s11219-014-9253-3

Svensson, R. B., Gorschek, T., Regnell, B., Torkar, R., Shahrokni, A., Feldt, R., & Aurum, A. (2011) Prioritization of quality requirements: State of practice in eleven companies. *2011 IEEE 19th International Requirements Engineering Conference*, 69-78. http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6051652

Sverrisdottir, H. S., Ingason, H. T., Jonasson, H. I. (2014). The role of the product owner in scrum – comparison between theory and practices. *Procedia Social and Behavioral Sciences*, 119(2014), 257-267. http://doi.org/10.1016/j.sbspro.2014.03.030

van Waardenburg, G., & van Vliet, H. (2013). When Agile meets the enterprise. *Information and Software Technology*, 55(12), 2154-2171. http://doi.org/10/1016/j.infsof.2013.07.012

Wood, S., Michaelides, G., & Thomson, C. (2013). Successful extreme programming: Fidelity to the methodology or good teamworking?. *Information and Software Technology*, 55(2013), 660-672. http://doi.org/10.1016/j.infsof.2016.10.002