SOFTENG 2025

The Eleventh International Conference on Advances and Trends in Software Engineering

ISBN: 978-1-68558-283-8

May 18 - 22, 2025

Nice, France

SOFTENG 2025 Editors

Luigi Lavazza, Università dell'Insubria - Varese, Italy

SOFTENG 2025

Forward

The Eleventh International Conference on Advances and Trends in Software Engineering (SOFTENG 2025), held between May 18-22, 2025 in Nice, France, continued a series of events focusing on the challenging aspects for software development and deployment, across the whole life-cycle.

Software engineering exhibits challenging dimensions in the light of new applications, devices and services. Mobility, user-centric development, smart-devices, e-services, ambient environments, e-health and wearable/implantable devices pose specific challenges for specifying software requirements and developing reliable and safe software. Specific software interfaces, agile organization and software dependability require particular approaches for software security, maintainability, and sustainability.

We welcomed academic, research and industry contributions. The conference had the following tracks:

- Challenges for dedicated software, platforms, and tools
- Software testing and validation
- Software requirements
- Maintenance and life-cycle management

We take here the opportunity to warmly thank all the members of the SOFTENG 2025 technical program committee, as well as all the reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and effort to contribute to SOFTENG 2025. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

We also thank the members of the SOFTENG 2025 organizing committee for their help in handling the logistics and for their work that made this professional meeting a success.

We hope that SOFTENG 2025 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in the field of software engineering. We also hope that Nice provided a pleasant environment during the conference and everyone saved some time to enjoy the historic charm of the city.

SOFTENG 2025 Chairs

SOFTENG Steering Committee
Zeeshan Ali Rana, NUCES, Lahore, Pakistan
Tsuyoshi Nakajima(中島毅), Shibaura Institute of Technology, Japan
Simona Vasilache, University of Tsukuba, Japan

SOFTENG Publicity Chairs
José Miguel Jiménez, Universitat Politecnica de Valencia, Spain
Francisco Javier Díaz Blasco, Universitat Politècnica de València, Spain
Ali Ahmad, Universitat Politècnica de València, Spain
Laura Garcia, Universidad Politécnica de Cartagena, Spain
Sandra Viciano Tudela, Universitat Politecnica de Valencia, Spain

SOFTENG 2025

Committee

SOFTENG Steering Committee

Zeeshan Ali Rana, NUCES, Lahore, Pakistan
Tsuyoshi Nakajima(中島毅), Shibaura Institute of Technology, Japan
Simona Vasilache, University of Tsukuba, Japan

SOFTENG 2025 Publicity Chairs

José Miguel Jiménez, Universitat Politecnica de Valencia, Spain
Francisco Javier Díaz Blasco, Universitat Politècnica de València, Spain
Ali Ahmad, Universitat Politècnica de València, Spain
Laura Garcia, Universidad Politécnica de Cartagena, Spain
Sandra Viciano Tudela, Universitat Politecnica de Valencia, Spain

SOFTENG 2025 Technical Program Committee

Khelil Abdelmajid, Landshut University of Applied Sciences, Germany
Mo Adda, University of Portsmouth, UK
Bestoun S. Ahmed, Karlstad University, Sweden
Issam Al-Azzoni, Al Ain University of Science and Technology, UAE
Vahid Alizadeh, College of Computing & Digital Media - DePaul University, USA
Washington Almeida, Cesar School | Center of Advanced Studies and Systems of Recife, Brazil
Vu Nguyen Huynh Anh, Université Catholique de Louvain, Belgium
Pablo O. Antonino, Fraunhofer IESE, Germany
Darlan Arruda, University of Western Ontario, Canada
Jocelyn Aubert, Luxembourg Institute of Science and Technology (LIST), Luxembourg
Heitor Augustus Xavier Costa, Federal University of Lavras (UFLA), Brazil
Lerina Aversano, University of Sannio, Italy
Ali Babar, University of Adelaide, Australia
Doo-Hwan Bae, Software Process Improvement Center - KAIST, South Korea
Mohamed Basel Almourad, College of Technological Innovation - Zayed University, Dubai, UAE
Bernhard Bauer, University of Augsburg, Germany
Imen Ben Mansour, University of Manouba, Tunisia
Maya Benabdelhafid, Ecole Supérieure de Comptabilité et de Finances (ESCF) de Constantine, Algeria
Marciele Berger, University of Minho, Portugal
Marcello M. Bersani, Politecnico di Milano, Italy
Anna Bobkowska, Gdansk University of Technology, Poland
Pierre Bourque, ETS Montreal, Canada
Fernando Brito e Abreu, ISCTE-IUL & ISTAR-IUL, Portugal
Antonio Brogi, University of Pisa, Italy
Azahara Camacho, RTI - Real Time Innovations, Spain
José Carlos Metrôlho, Polytechnic Institute of Castelo Branco, Portugal

Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission or reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article is does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

Table of Contents

Beyond Code: The PPPT Framework for Holistic Software Success 1
Vivek Jain

Putting Business Goals in Context for Measurement 8
Luigi Lavazza, Sandro Morasca, and Davide Tosi

Beyond Code: The PPPT Framework for Holistic Software Success

Vivek Jain
Software Development Manager
Academy Sports plus Outdoors
Texas, USA
e-mail: vivek65vinu@gmail.com

***Abstract*— The software industry is continuously evolving, requiring robust frameworks to manage innovation, efficiency, and scalability. The People, Processes, Products, and Technology (PPPT) framework provides a holistic approach to software development and management. This paper explores the PPPT framework in detail, discussing its significance, implementation strategies, and real-world case studies that illustrate its effectiveness. By synthesizing existing research and industry insights, we provide a comprehensive understanding of how organizations can leverage this framework to achieve competitive advantages.**

Keywords—PPPT Framework; Software Industry; Digital Transformation; Agile Methodology; Operational Efficiency.

I. INTRODUCTION

The rapid evolution of the software industry has necessitated frameworks that integrate diverse aspects of operations [1][2][3]. The PPPT framework, originally conceptualized for business management [4][5], has gained traction in software engineering to balance technical and human-centric objectives. While individual components such as agile methodologies [6], product management [7], or technological advancements [8] have been explored extensively, their unified impact within the PPPT framework remains under-researched. This paper aims to:

1) Define the PPPT framework in the context of software development.

2) Analyze the synergy between people, processes, products, and technology.

3) Propose implementation strategies and metrics to evaluate success.

II. THE PPPT FRAMEWORK: STRUCTURE AND INTERDEPENDENCE

In this section, we shall explain the PPPT framework.

A. Definition of the PPPT Framework

The PPPT framework is a comprehensive and structured model that integrates four foundational pillars—People, Processes, Products, and Technology—to enhance the efficiency, scalability, and innovation capacity of software organizations [9]. It acts as a blueprint for aligning talent, operations, product strategy, and technological adoption, with the goal of delivering high-impact digital solutions.

B. Purpose and Strategic Importance

The framework serves as a strategic enabler for software-driven enterprises navigating complex, fast-paced environments [10]. It encourages:

1) Cross-functional collaboration by aligning people and processes.

2) Operational efficiency through standardized workflows.

3) Innovation and user satisfaction via product-driven strategies.

4) Competitive advantage through timely adoption of emerging technologies [11][12].

By breaking down silos between departments and linking technology decisions directly to user needs and business outcomes, the PPPT framework helps create a unified and responsive development ecosystem [13][14].

C. Interconnected Components and Their Roles

The strength of the PPPT framework lies in the synergy between its components, not just their individual contributions.

1) People: People are the central drivers of value in software development. Their skills, mindset, and collaboration impact all other components. For example, product innovation (Products) thrives when empowered teams (People) work within adaptive development frameworks (Processes) and leverage enabling tools (Technology) [1] [2][15].

2) Processes: Processes act as the connective tissue that links people to outcomes. Agile, DevOps, and CI/CD pipelines formalize collaboration and drive iterative value delivery [6][16][17]. Well-defined processes reduce friction, enabling teams to respond quickly to changes in product strategy or technological shifts.

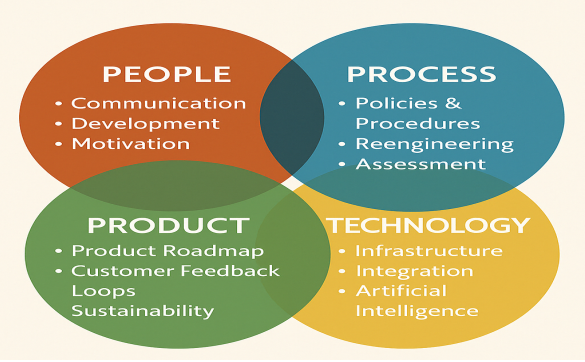

Figure 1. The PPPT framework shows how People, Processes, Products, and Technology interact in a dynamic, mutually reinforcing ecosystem

3) Products: Products represent the tangible outcomes of the collective effort. The feedback loop between product usage and development teams (People) shapes process refinements and technology upgrades [18][19]. Lean Startup principles and customer journey analytics ensure the product evolves in lockstep with user expectations [37][20].

4) Technology: Technology provides the tools and platforms that empower people, automate processes, and bring product visions to life. Whether it's cloud infrastructure or AI-driven analytics, the chosen tech stack must support and accelerate the other three pillars [12][14][34].

D. Visualizing the Interdependencies

Figure 1 illustrates how the four core elements interconnect dynamically. Each component influences and is influenced by the others. For example, changes in technology (e.g., adopting Kubernetes [31]) may necessitate new processes (CI/CD automation), which in turn reshape team responsibilities (People) and impact the delivery model (Products).

E. Instantiating the Framework

To operationalize the PPPT framework, organizations must go beyond conceptual understanding and implement targeted actions that map each component to practical workflows and measurable outcomes.

1) People: Start by conducting role-mapping workshops to ensure that team responsibilities align with project objectives and process flows. Foster a culture of continuous learning through skill development programs and leadership coaching [25][26]. Encourage cross-functional collaboration by forming agile squads that include members from product, engineering, Quality Assurance (QA), and User Experience (UX) disciplines.

2) Processes: Establish a process maturity model to assess current workflows and identify gaps. Introduce Agile or SAFe methodologies, depending on the organization's scale and readiness [5][27][28]. Incorporate regular retrospectives, sprint reviews, and continuous integration pipelines to reinforce adaptability [31][33]. Use value stream mapping to remove process bottlenecks and improve end-to-end visibility.

3) Products: Apply customer-centric design approaches such as Design Thinking, Lean User Experience (UX), and A/B testing [37] to validate assumptions and iterate on product features. Create product roadmaps that reflect not only technical feasibility but also user feedback and market trends [13][38]. Align product key performance indicators (KPIs) with strategic business goals to measure real impact.

4) Technology: Perform technology assessments to ensure your stack supports long-term scalability and agility. Introduce modern tooling such as GitOps, Infrastructure as Code (IaC), or cloud-native platforms to support continuous delivery [31][34]. Ensure that selected technologies integrate seamlessly with your processes and enhance—not hinder—team productivity [35].

5) Feedback and Metrics Integration: Successful instantiation also includes setting up closed-loop feedback mechanisms that span all four pillars. This could involve regular employee engagement surveys (People), DORA metrics for DevOps efficiency (Processes) [33], NPS and churn rates (Products), and cost-performance dashboards for cloud resources (Technology) [18][19][21].

6) Governance and Adaptability: Finally, establish a lightweight governance layer to monitor alignment, maintain standards, and allow for controlled experimentation. The framework should be flexible enough to evolve as the organization grows or pivots [39][40], ensuring it remains relevant in dynamic market conditions.

By approaching implementation through an iterative, evidence-based lens, organizations can transform the PPPT framework from a theoretical model into a living system that drives continuous innovation and sustained value delivery.

III. LITERATURE REVIEW

The evolution of frameworks in the software industry has been driven by the need to manage complexity, ensure quality, and deliver value to stakeholders. Numerous studies have explored the influence of people, processes, products, and technology independently or in partial combination. However, a comprehensive framework that integrates all four pillars—People, Processes, Products, and Technology (PPPT)—remains underdeveloped in both academic literature and industrial practice [9][10].

A. People-Centric Approaches

Human factors such as team dynamics, leadership, skillsets, and communication have been widely studied. For instance, DeMarco and Lister's seminal work *Peopleware* [1] emphasizes the significance of a conducive work culture and managerial support. Further, studies like those by Lenberg et al. [2] and Schein [26] stress psychological and social dynamics in software teams, yet they do not establish a systemic framework to connect these factors with process or technology outcomes.

B. Process Improvement Models

Traditional process-oriented frameworks, such as CMMI [3], ITIL [4], and Agile methodologies [5][6][22] have prioritized repeatability and efficiency. While these models provide robust process control mechanisms, they often lack integration with human aspects and rapidly evolving technology. Agile addresses people and processes [29] but largely omits structured considerations of product innovation and technological evolution in a unified manner.

C. Product Lifecycle and Innovation Models

Product-centric studies often focus on development lifecycle management, usability, and market responsiveness. Boehm's Spiral Model [6] and V-Model approaches highlight risk mitigation and quality control. Additional studies [13][19][38] emphasize product strategy and market dynamics. However, these models typically treat product development in isolation from organizational people dynamics or broader technological adaptation strategies.

D. Technology Adaptation Frameworks

Technology integration frameworks such as TOGAF [7] and DevOps pipelines [30][31][34] have garnered significant attention for enabling automation, scalability, and speed.

Nonetheless, these models are technology-heavy and often overlook the cultural and organizational readiness aspects, particularly in cross-functional teams.

E. Integrated Frameworks and Emerging Efforts

Recent efforts like the socio-technical systems theory [8] and the Leavitt Diamond Model [9], and digital transformation approaches from Gartner [17], McKinsey [18], Deloitte [19], and IBM [21] attempt to link organizational elements together. However, these models either remain too generic for software industry application or lack actionable strategies for continuous improvement across all PPPT components. Moreover, there is limited empirical validation or adaptation of such frameworks within dynamic software delivery environments [12][23][24][39].

F. Identified Research Gap

While literature abundantly discusses each component—People, Processes, Products, and Technology—in isolation or in pairs (e.g., Agile's focus on People and Process [5][15]; DevOps on Process and Technology [31][34]), an integrated, practical, and adaptable framework that holistically captures the interdependencies among all four pillars remains conspicuously absent. This research seeks to fill that gap by proposing the PPPT Framework, a holistic model designed specifically for the modern software industry. It aims to guide organizations in aligning human factors, process maturity, product strategy, and technological innovation for sustained success [40].

IV. Framework In Action: Interdependencies And Implementation

The real power of the People, Processes, Products, and Technology (PPPT) framework lies in how its components interlock and reinforce one another. Successful adoption isn't just about understanding each pillar, but also about orchestrating their synergy and translating it into measurable action.

A. Interconnected Pillars: Driving Synergy

1) People ↔ Process: The interaction between people and processes are the lifeblood of organizational efficiency. Collaboration tools like Jira, Microsoft Teams, and Slack [36] support agile workflows, foster transparency, and improve process adherence, helping teams stay aligned on goals and execution.

2) Process ↔ Technology: Technology isn't just a support mechanism—it's a strategic enabler. Tools such as CI/CD pipelines, automated testing frameworks, and DevOps toolchains [31][34] streamline processes, reduce human error, and enforce consistency, making process execution fast and reliable.

3) Product ↔ Technology/Process: Product evolution is often shaped by customer needs and feedback. This demand feeds back into technology choices (e.g., adopting microservices or AI/ML) [18][19] and necessitates process enhancements (e.g., faster iteration cycles or more robust Quality Assurance (QA)). In this way, product direction becomes a catalyst for continuous improvement across the framework.

B. Implementation Strategies: Turning Theory into Practice

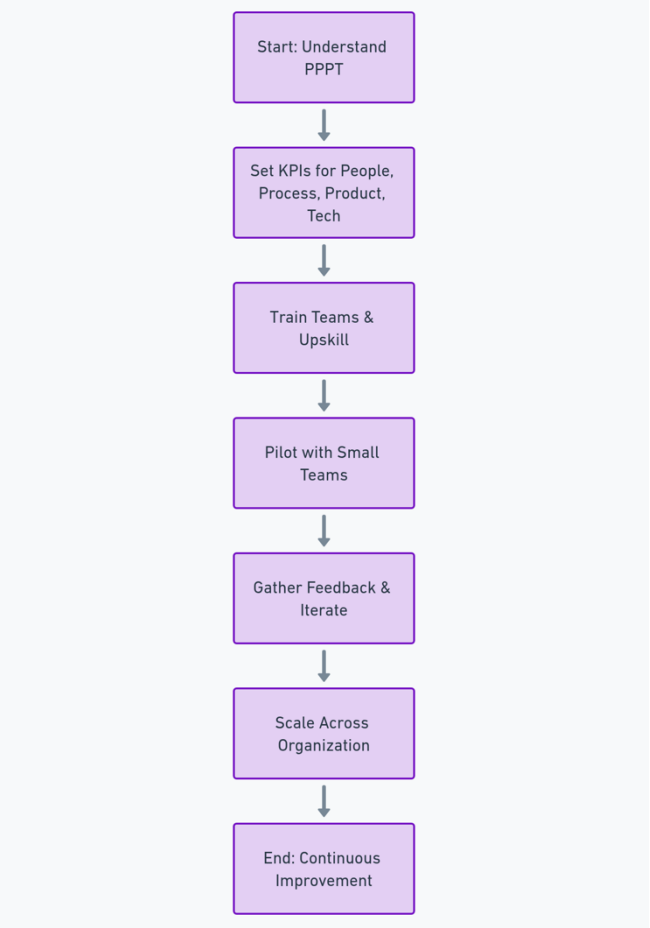

Figure 2. PPPT Framework Operationalization Flow

Putting the PPPT framework into motion requires both strategic intent and operational discipline:

1) Establish Key Performance Indicators (KPIs) That Matter: Define performance metrics aligned to each pillar. Some examples:

a) People: Team satisfaction, collaboration effectiveness

b) Process: Sprint velocity, schedule adherence, defect density

c) Product: Feature adoption, customer satisfaction (CSAT, NPS)

d) Technology: Deployment frequency, incident resolution time, first-time pass rate

2) Prioritize Training and Upskilling: Equip teams with knowledge through Agile, DevOps, and tech enablement workshops [5][25][28][31]. This not only aligns teams with business goals but also fuels innovation.

3) Pilot and Iterate: Start small—use a few cross-functional teams to test the integration of the framework. Use insights from these pilots to refine your approach before scaling across departments or the organization.

4) Feedback Loops: Create structured mechanisms (retrospectives, product feedback sessions, review boards) to gather insights and refine the framework continuously.

V. CASE STUDIES

The case studies presented were selected based on purposive sampling, focusing on organizations that explicitly adopted components of the PPPT framework over a 12–24 month period. Data sources included publicly available transformation reports, interviews with stakeholders (where available), and published metrics from internal dashboards. Each case was analyzed by mapping initiatives to the four PPPT dimensions (People, Processes, Products, Technology), followed by outcome tracking across 3–5 measurable key performance indicators (KPIs) such as release velocity, customer satisfaction, and operational cost efficiency. This ensured a consistent and structured comparison of PPPT implementation effectiveness.

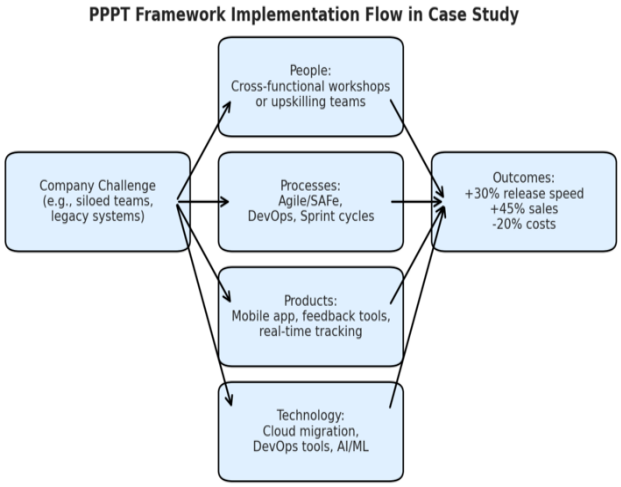

Figure 3. Visual Flow of PPPT Framework Implementation in Case Studies

In this section, we present two real-world case studies that demonstrate how organizations implemented the PPPT framework to overcome digital transformation challenges. Each study includes the organization's background, framework-driven interventions, and measurable outcomes.

A. Company A: Agile at Scale (SaaS Sector)

1) Background: Company A, a global SaaS provider, struggled with managing distributed teams across multiple regions. These silos caused inefficiencies, delayed releases, and inconsistent customer experiences.

2) PPPT Framework Approach: Adopting the PPPT framework, the organization emphasized:

a) People: Conducted cross-functional team workshops to improve collaboration and understanding of shared objectives.

b) Processes: Implemented the Scaled Agile Framework (SAFe) to align development across distributed teams.

c) Products: Integrated continuous feedback mechanisms through user surveys and analytics tools.

d) Technology: Leveraged cloud-based DevOps tools like Azure DevOps and Kubernetes [31][19] to streamline deployments.

3) Outcomes:

a) Release cycles were reduced by 30%.

b) Customer satisfaction improved due to faster feature delivery.

c) Employee engagement increased as teams felt more empowered and aligned.

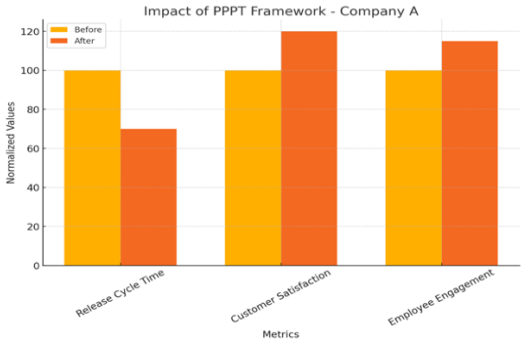

Figure 4. Impact of PPPT Framework – Company A

B. Company B: Digital Transformation in Retail

1) Background: Company B, a major retail chain, struggled to create a seamless omnichannel shopping experience. Legacy systems were a bottleneck, and customer feedback loops were fragmented.

2) PPPT Framework Approach: By adopting the PPPT framework, the company achieved transformation through:

a) People: Upskilled IT and marketing teams to use analytics tools and AI-driven insights [12][18].

b) Processes: Adopted DevOps practices and transitioned to Agile sprint cycles for iterative development.

c) Products: Launched an integrated mobile app with real-time inventory tracking and personalized recommendations.

d) Technology: Migrated core systems to the cloud and implemented AI algorithms for customer segmentation and recommendation [13].

3) Outcomes:

a) Online sales grew by 45% within the first year.

b) Customer retention rates increased due to improved personalized shopping experiences.

c) Operational costs decreased by 20% due to automation and improved inventory management.

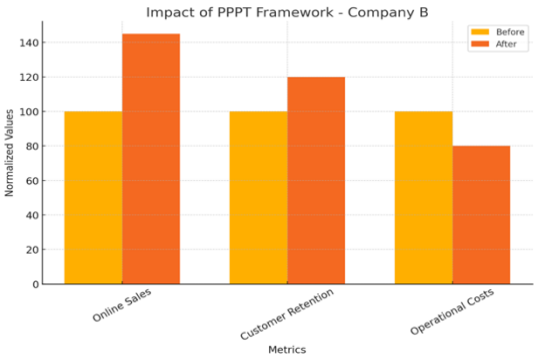

Figure 5. Impact of PPPT Framework – Company B

VI. Challenges And Solutions

In this section, we present challenges and solutions. Organizations implementing the PPPT framework face various practical hurdles. Many of these challenges have been documented in change management literature [24][26][35] and highlight the importance of strong leadership, communication, and adaptability [40].

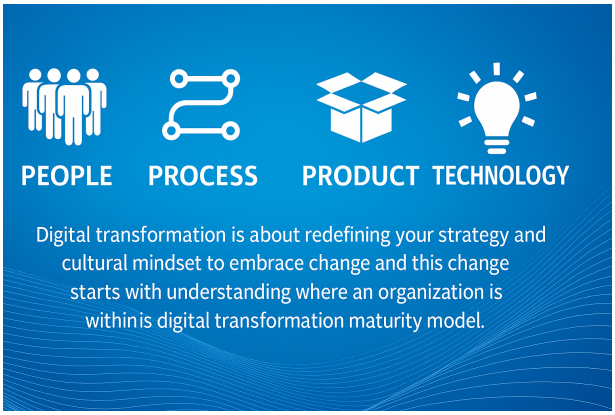

Figure 6. Digital Transformation: People, Process, Product & Technology

A. Resistance to Change

1) Challenge: People often resist change due to fear of the unknown, perceived threats to job security, or skepticism about the framework's effectiveness [1][26]. This resistance is particularly prevalent in organizations with a long history of traditional practices.

2) Solution:

a) Change Management Programs: Create a structured approach for managing change, including clear communication about the benefits of the PPPT framework and its alignment with organizational goals.

b) Stakeholder Involvement: Engage employees at all levels during the planning and execution stages to foster a sense of ownership and reduce resistance.

c) Training and Upskilling: Offer workshops, certifications, and practical training to help employees adapt to new technologies and processes.

B. Silos Between Teams: In large organizations, teams often work in isolation, leading to misaligned goals, duplicated efforts, and inefficiencies. This is a well-known barrier addressed in agile and DevOps literature [6][31][35].

1) Challenge: In large organizations, teams often work in isolation, leading to misaligned goals, duplicated efforts, and inefficiencies. This is particularly problematic when integrating diverse components like people, processes, products, and technology.

2) Solution:

a) Cross-Functional Teams: Establish collaborative teams that include representatives from development, operations, product management, and customer support.

b) Unified Communication Platforms: Use tools like Slack, Microsoft Teams, or Jira to facilitate real-time collaboration and transparency.

c) Shared Key Performance Indicators (KPIs): Define performance indicators that align with organizational objectives and promote inter-team accountability.

C. Measuring Intangibles

1) Challenge: It is difficult to measure abstract elements like collaboration, innovation, and user satisfaction within the PPPT framework. Hybrid approaches involving metrics like NPS, DORA, and employee engagement surveys are widely recommended [20][33][34]. Without clear metrics, organizations struggle to evaluate success and justify investments.

2) Solution:

a) Hybrid Metrics: Combine qualitative methods (e.g., employee surveys, customer interviews) with quantitative measures (e.g., Net Promoter Score, defect rates, and velocity metrics).

b) AI and Analytics: Leverage machine learning and analytics tools to track patterns in team collaboration and customer interactions.

c) Iterative Evaluation: Use regular retrospectives and checkpoints to adjust metrics based on evolving goals.

D. Legacy Systems and Technology Debt: Legacy systems can hinder digital transformation efforts. Approaches such as microservices architecture and incremental modernization are outlined in architecture best practices [14][27][30].

1) Challenge: Many organizations face significant hurdles due to outdated systems that lack integration capabilities and require expensive maintenance, impeding technology adoption.

2) Solution:

a) Gradual Migration: Transition to modern systems incrementally to minimize operational disruptions.

b) Microservices Architecture: Adopt modular systems to replace monolithic legacy applications, enabling scalability and easier updates.

c) Investment in DevOps: Automate testing and deployment processes to streamline updates to legacy systems during migration.

E. Balancing Innovation with Operational Stability: Frameworks like Bimodal IT and innovation sandboxes help manage this balance [19][32][39].

1) Challenge: Pursuing innovation often conflicts with the need to maintain operational stability. Organizations might find themselves prioritizing one at the expense of the other.

2) Solution:

a) Dual Operating Models: Use a bimodal approach where one team focuses on innovation (Mode 2) while another ensures operational excellence (Mode 1).

b) Sandbox Environments: Create isolated environments for experimenting with new technologies without risking core operations.

c) Risk Mitigation Plans: Develop strategies to manage risks associated with deploying innovative features, including rollbacks and phased launches.

F. Budget and Resource Constraints: Cost-effective adoption of digital strategies has been emphasized in transformation reports [17][18][21].

1) Challenge: Adopting the PPPT framework requires

significant upfront investment in training, tools, and restructuring, which may strain limited budgets.

2) Solution:

a) Phased Implementation: Prioritize initiatives with high ROI and implement the framework in stages to manage costs.

b) Vendor Partnerships: Collaborate with technology vendors to access cost-effective solutions, training, and support.

c) Grants and Incentives: Seek government grants or industry programs that support digital transformation and innovation.

G. Cultural Misalignment: Organizational culture significantly influences the success of any transformation [8], [9][25].

1) Challenge: Organizations with rigid hierarchical cultures may struggle to adapt to the collaborative and iterative nature of the PPPT framework.

2) Solution:

a) Cultural Transformation Initiatives: Encourage openness, experimentation, and continuous learning through leadership advocacy and reward systems.

b) Leadership Buy-In: Ensure executives model the desired behaviors and actively support PPPT adoption.

c) Feedback Loops: Build a culture of transparency by continuously gathering and acting on employee and customer feedback.

By addressing these challenges systematically, organizations can increase the likelihood of successful PPPT framework implementation, fostering sustainable growth and innovation

VII. Conclusion and Future work

In this paper, we introduced the PPPT (People, Processes, Products, and Technology) Framework as a holistic model to address the multifaceted challenges in the software industry. While existing literature and practices often focus on isolated dimensions—such as process improvement or technological advancement—the PPPT Framework emphasizes the interconnectedness of human factors, operational methodologies, product strategies, and emerging technologies [10][18][40].

Through a detailed review of related work, we identified a significant research gap: the lack of an integrated framework that supports continuous alignment across all four critical dimensions. Our proposed framework aims to fill this void by offering organizations a systematic approach to digital transformation, operational efficiency, and product innovation.

The PPPT Framework serves not only as a conceptual guide but also as a practical tool for software organizations seeking to adapt in a rapidly evolving ecosystem. It fosters a balanced perspective that supports sustainable growth, improved collaboration, and agility in response to market demands.

Future studies ought to concentrate on a few crucial areas: As technology continues to evolve, the PPPT framework must adapt to emerging trends such as artificial intelligence, quantum computing, and decentralized applications [12][19]. Future research should focus on integrating AI-driven decision-making processes into the framework to enhance predictive analytics and automation [11][34]. Additionally, exploring cross-industry applications of the PPPT framework in fields such as healthcare, finance, and manufacturing can offer new insights into its versatility [13][39]. Organizations should also examine sustainability within the PPPT framework, ensuring that technology and product development align with environmental and ethical standards [16][37]. By continuously refining and expanding the PPPT framework, businesses can maintain resilience and adaptability in an ever-changing digital landscape.

References

[1] T. DeMarco and T. Lister, *Peopleware: Productive Projects and Teams*, 2nd ed., Dorset House, 1999.

[2] P. Lenberg, R. Feldt, and L. G. Wallgren, "Behavioral software engineering: A definition and systematic literature review," *J. Syst. Softw.*, vol. 107, pp. 15–37, Jun. 2015.

[3] SEI, "Capability Maturity Model Integration (CMMI) for Development," Version 1.3, 2010.

[4] Office of Government Commerce (OGC), *ITIL Lifecycle Suite*, TSO, 2011.

[5] K. Beck et al., *Manifesto for Agile Software Development*, 2001.

[6] B. Boehm, "A spiral model of software development and enhancement," *Computer*, vol. 21, no. 5, pp. 61–72, 1988.

[7] The Open Group, *TOGAF® Version 9.2*, The Open Group, 2018.

[8] E. Trist, "The evolution of socio-technical systems," *Occasional paper*, vol. 2, 1981.

[9] H. J. Leavitt, "Applied organization change in industry: Structural, technological and humanistic approaches," in *New Perspectives in Organizational Research*, Wiley, 1965.

[10] T. H. Davenport and J. E. Short, "The new industrial engineering: Information technology and business process redesign," *Sloan Manage. Rev.*, vol. 31, no. 4, pp. 11–27, 1990.

[11] B. Fitzgerald and K. J. Stol, "Continuous software engineering: A roadmap and agenda," *J. Syst. Softw.*, vol. 123, pp. 176–189, 2017.

[12] A. McAfee, E. Brynjolfsson, T. H. Davenport, D. J. Patil, and D. Barton, "Big data: The management revolution," *Harvard Bus. Rev.*, vol. 90, no. 10, pp. 60–68, 2012.

[13] M. A. Cusumano, S. J. Kahl, and F. F. Suarez, "Services, industry evolution, and the competitive strategies of product firms," *Strateg. Manage. J.*, vol. 36, no. 4, pp. 559–575, 2015.

[14] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice*, 3rd ed. Boston, MA, USA: Addison-Wesley, 2012.

[15] K. Schwaber and J. Sutherland, *The Scrum Guide*, 2020. [Online]. Available: https://scrumguides.org

[16] M. E. Porter, *Competitive Advantage: Creating and Sustaining Superior Performance*. New York, NY, USA: Free Press, 1985.

[17] Gartner, *Digital Business Transformation: A Gartner Trend Insight Report*, 2020.

[18] McKinsey & Company, *How Software Excellence Fuels Business Performance*, 2022. [Online]. Available: https://www.mckinsey.com

[19] Deloitte, *Tech Trends 2021: The Disruptors*, 2021. [Online]. Available: https://www2.deloitte.com

[20] Forrester Research, *The State of Agile Development*, 2020.

[21] IBM Institute for Business Value, *The Enterprise Guide to Digital Transformation*, 2020.

[22] B. Boehm and R. Turner, "Balancing agility and discipline: A guide for the perplexed," in *Proc. Int. Conf. Softw. Eng. (ICSE)*, 2004.

[23] H. Kerzner, *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*. Hoboken, NJ, USA: Wiley, 2013.

[24] M. Hammer and J. Champy, *Reengineering the Corporation: A Manifesto for Business Revolution*. New York, NY, USA: Harper Business, 1993.

[25] J. A. Highsmith, *Agile Software Development Ecosystems*. Boston, MA, USA: Addison-Wesley, 2002.

[26] E. H. Schein, *Organizational Culture and Leadership*. San Francisco, CA, USA: Jossey-Bass, 2010.
[27] I. Sommerville, *Software Engineering*, 10th ed. Boston, MA, USA: Pearson, 2016.
[28] J. A. Highsmith, *Agile Project Management: Creating Innovative Products*. Boston, MA, USA: Addison-Wesley, 2009.
[29] M. Larman, *Scaling Lean & Agile Development*. Boston, MA, USA: Addison-Wesley, 2008.
[30] F. Brooks, *The Mythical Man-Month: Essays on Software Engineering*. Boston, MA, USA: Addison-Wesley, 1995.
[31] L. Bass, I. Weber, and L. Zhu, *DevOps: A Software Architect's Perspective*. Boston, MA, USA: Addison-Wesley, 2015.
[32] A. Brown, K. Hummel, and H. Reijers, *Agile Processes in Software Engineering and Extreme Programming*. Cham, Switzerland: Springer, 2019.
[33] B. Fitzgerald and K. J. Stol, "Continuous software engineering: A roadmap and agenda," *J. Syst. Softw.*, vol. 123, pp. 176–189, 2017.
[34] J. Humble, G. Kim, and N. Forsgren, *Accelerate: The Science of Lean Software and DevOps*. Portland, OR, USA: IT Revolution, 2018.
[35] G. Kim, K. Behr, and G. Spafford, *The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win*. Portland, OR, USA: IT Revolution, 2013.
[36] H. Kniberg, *Spotify Engineering Culture*, 2012. [Online]. Available: https://spotify.github.io
[37] E. Ries, *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. New York, NY, USA: Crown Business, 2011.
[38] J. Sutherland, *Scrum: The Art of Doing Twice the Work in Half the Time*. New York, NY, USA: Crown Business, 2014.
[39] G. Westerman, D. Bonnet, and A. McAfee, *Leading Digital: Turning Technology into Business Transformation*. Boston, MA, USA: Harvard Business Review Press, 2014.
[40] V. Jain, "People, processes, products and technology (PPPT) framework in the software industry," *J. Artif. Intell. Cloud Comput.*, vol. 3, 2024, doi: 10.47363/JAICC/2024(3)E244.

 ISBN: 978-1-68558-283-8

Putting Business Goals in Context for Measurement

Luigi Lavazza
Dip. di Scienze Teoriche e Applicate
Università degli Studi dell'Insubria
Varese, Italy
email:luigi.lavazza@uninsubria.it

Sandro Morasca
Dip. di Scienze Teoriche e Applicate
Università degli Studi dell'Insubria
Varese, Italy
email:sandro.morasca@uninsubria.it

Davide Tosi
Dip. di Scienze Teoriche e Applicate
Università degli Studi dell'Insubria
Varese, Italy
email:davide.tosi@uninsubria.it

***Abstract*—Effective software measurement in a business organization requires a deep understanding of the business context, i.e., the business world in which the organization operates. Thus, there is a need for describing the business world and placing business goals into their context, so sensible measurement plans can be defined and enacted. In this paper, based on Jackson's ideas on domain representation and using concepts from the GQM+Strategies technique, we propose a method to precisely describe the business domain and its characteristics, the business goals, the strategies, their relationships with the software activities carried out to support the strategies, and how strategies are selected. Specifically, we propose a way to describe the business world first, including business and software processes, and then specify the required measurements.**

Keywords–Software development process; Software process measurement; GQM (Goal/Question/Metric); Domain representation

I. Introduction

Business organizations need measures, to evaluate the performance of their processes, identify improvements, evaluate the effectiveness of changes, etc. Since business is largely supported by software, both business and IT people are interested in measuring how effective and efficient software is in supporting business.

The *business world* (BW) is the part of the real world relevant for the business, e.g., the market, users, stakeholders, competition, etc.. In the BW, *business goals* are conceived and *strategies* are deployed to achieve such goals. Goals and strategies are hierarchical in nature: implementing a strategy usually involves achieving a lower level goal, which, in turn, could require a strategy. A clear understanding of the business domain, the rules and constraints affecting the business, the final goals of the stakeholders, and the cause-effect relationships that govern the business is of fundamental importance to devise effective strategies. Those who need to support such strategies by means of software and then measure the effectiveness of the software solutions and the implemented strategies need to have access to explicit and clear descriptions of the BW. They have to distinguish between what is given (the context), what is currently not true and must be achieved (the business goal), and what is the set of actions (the strategy) that have been planned to achieve the business goal. In general, the context accounts for several elements (e.g., laws, standards, the market, etc.) that cannot be changed, at least within the considered project or activity; nonetheless, some parts of the context can be modified as needed (e.g., we can instruct an employee to perform some action that was not carried out previously).

The *measurement world* (MW) is where measurement plans are specified, measures are defined, and indicators (e.g., key process indicators) are computed. The MW is much more controllable than the BW, so, techniques and tools—like the Goal/Question/Metric (GQM) [1]–[3] and related tools and methodologies—have been defined to support the work to be carried out in the MW. Measurement should be used to assess strategies and goals at different hierarchical levels in the given business context, i.e., in the BW. Thus, there needs to be a two-way set of relationships between the BW and the MW. The BW contains the relevant objects of study, so the data used in the MW come from observations on the BW. Conversely, the MW needs to feed back to the BW the results of measurement and modeling activities. Thus, people from the two worlds need to at least agree on the measure definitions, how measurement is carried out, the meaning and expressiveness of indicators, etc.

GQM+Strategies [4] [5] highlights the relations existing between business goals and software development (or acquisition) within the BW and supports identifying and documenting the relationships between goals in the BW and measurement plans in the MW, as shown in Figure 1.

We here propose an approach to precisely describing the BW, in terms of the business domain characteristics, the business goals, the strategies and their relationships with the software activities that support the strategies. We argue that the elements used to select a strategy in a set of alternatives need to be explicit. Specifically, we recommend that the specific figure of merit and preference criterion among alternative strategies be made explicit and recorded, to evaluate the usefulness of the selection process, so it can be used in future strategy selections. Our proposal is based on ideas from Jackson's work [6]–[8] on requirements and domain representation, and uses concepts that have been formalized in GQM+Strategies.

The paper is organized as follows. Section II introduces a case study to explain our ideas throughout the paper. Section III concisely discusses the need for better (measurable) business models. Section IV proposes a (meta)model to represent the hierarchy of requirements in the BW; Section V discusses the selection of the best strategy; Section VI links business elements to measurement plans. Section VII accounts

for related work. We conclude and we draw some directions of investigation in Section VIII.

II. A Case Study

We use an example (taken from [9]) to illustrate our approach. A company operates in a market that is becoming highly competitive, so there is a need to safeguard the company's place in the market, i.e., to keep existing customers. To this end, generating customer loyalty is necessary. This can be achieved by improving customer satisfaction with the next product, so business goal "increase customer satisfaction by 10%" is defined.

An analysis revealed that many customer complaints are due to product reliability problems. After considering several possible strategies, it was decided that the most promising way to increase customers' satisfaction is to "test reliability in." Thus, the software test processes are examined and potential lower-level goals are identified. The company has found a new system test process that seems appropriate for the context and can decrease the total number of customer complaints by 10% by reducing customer-reported software field defects (i.e., those that slip by system test) by 20%. So, a second-level goal, "improve system test effectiveness by 20%," can be defined. Because there is a new suitable system test process, the only strategy available is to introduce the new system test process.

Based on historical defect slippage data, the company assumes that reducing slippage by 20% reduces reported defects by 20%. So, the lower level goal is to apply the new system test method to see if it actually reduces defect slippage by at least 20% and eventually generates the necessary improvement to customer complaints.

III. On the need for better models of BW

Consistent with GQM+Strategies, in our approach we specify *Business goals* in a *Context* (some of whose characteristics are known with certainty, while others are represented by *Assumptions*) and devising *Strategies* to reach the *Business goals*.

First, the boundaries of the BW model should be explicitly defined. Similarly, it should be clarified why some elements of the BW are in the model, while others have been excluded. Given a business goal, it is always possible to wonder from where it originates, what business needs led to the definition of such goal, etc. However, the specific problem to which top level context and assumptions (namely: the market is competitive; customer satisfaction increases customer loyalty) are related is not mentioned, so we do not know if there is an even higher-level goal that can be reached by pursuing the example's top-level goal. One might infer that the (unknown) higher-level business goal is to increase customers' loyalty, or just to preserve the current market share, since in a competitive market, improving customers' satisfaction could be necessary to just preserve company's market share. In general, there may be an upward chain of several goals, so the top-level goal should be given as an "axiom," and no further context or assumptions should be provided to justify it.

At the opposite end, a goal that is at the ground level in a model can always call for a strategy. In fact, any goal that can be pursued in two or more different ways can be associated with a "strategy" that indicates which of the several possible implementation ways has been chosen.

Let us now consider the fact that several different strategies can possibly satisfy a given business goal. For instance, customers' satisfaction can be increased in several different manners: increasing the reliability of products is surely a way, but it could be possible to decrease prices, add functions, improve efficiency, etc. Explicitly recording the decision criteria that lead to selecting a strategy would be beneficial, since decision criteria could play a very important role in the evaluation of strategies. Over time, by recording the decisions made, their rationales, and the results obtained, we can reach a reliable evaluation of the strategy selection criterion that can be recorded (e.g., in an Experience Factory [10]) as an asset of the organization for future use.

IV. Describing Requirements Hierarchies

Given a context and a goal, the strategy is the "solution" that—in the given context and under the given assumptions—satisfies the goal. Using Jackson's concepts and notation [6], the statement above can be written as follows

$$Context, Strategy \vdash Goal \tag{1}$$

where *Context* is the description of the business domain, including all knowledge relevant to the goals currently considered in the form of known facts or assumptions, *Goal* is the description of what is desired by the business actors, and *Strategy* is the solution devised to achieve the *Goal*.

The logical entailment $A \vdash B$ states that from assuming A we can prove B. The level of formality of Formula (1) depends on the formality of *Context*, *Strategy* and *Goal*: if they are described formally, it is possible to prove that the achievement of *Goal* descends from the statements in *Context* and *Strategy* being true. Instead, informal descriptions allow only for argumentations, which are however deemed sufficient in most cases.

In Jackson's terminology, the context is given, thus it is "indicative." However, part of the context can sometimes be controlled or changed: this part of the context is therefore not indicative. Actually, changing it could be part of a strategy.

The Goal is "optative," i.e., it represents something that is not currently true, but needs to be made true by applying the Strategy in the Context.

The Strategy is clearly optative, since in general the Goal can be achieved via several different strategies. Once a Strategy has been described, i.e., we have decided *what* has to be achieved, it is necessary to specify *how* it should be achieved. Thus, goals and strategies form hierarchies [11]: implementing a strategy in general requires the achievement of some lower-level goal, which calls for a lower-level strategy, which could require the achievement of an even lower-level goal, etc. This

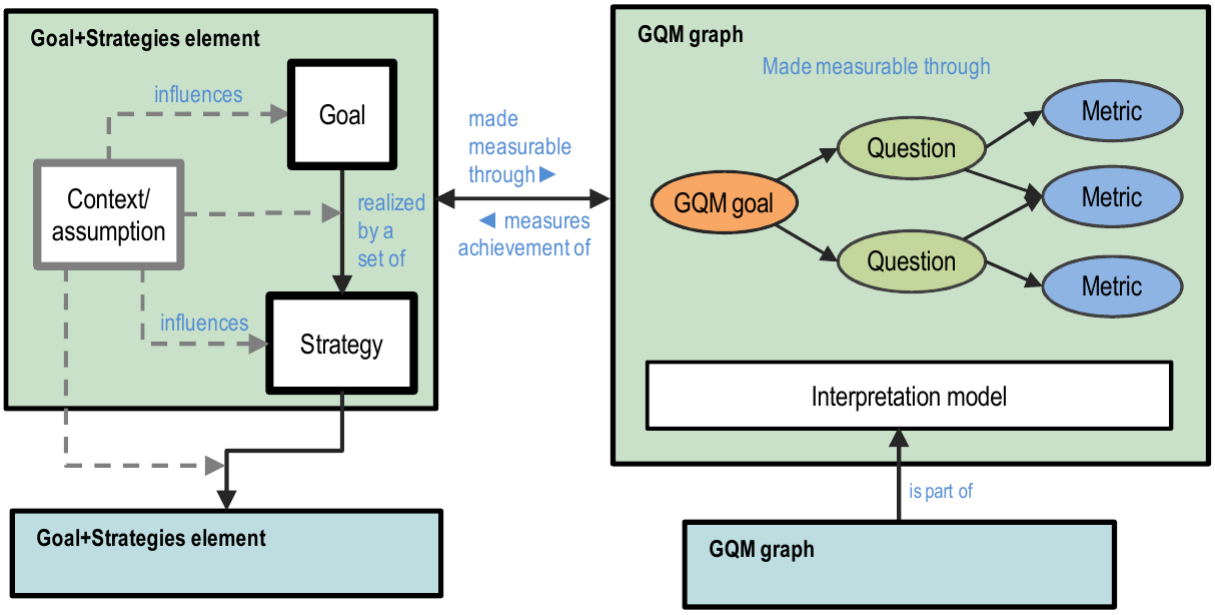

Fig. 1. GQM+strategies.

type of hierarchies can be described using Jackson's notation as follows:

$$Context, Strategy \vdash BusinessGoal$$
$$Context, LowerLevelGoal \vdash Strategy$$
$$Context, LowerLevelStrategy \vdash LowerLevelGoal$$

The LowerLevelGoal specifies *what* we can do to realize the Strategy. Reaching LowerLevelGoal in the Context is a sufficient condition for the realization of the Strategy. However, LowerLevelGoal is a goal, so it is again necessary to specify how LowerLevelGoal itself should be achieved. To this end, we need to devise a LowerLevelStrategy to reach LowerLevelGoal as shown in the last logical entailment above.

Note that in formula $Context, Strategy \vdash BusinessGoal$ the Strategy is optative, while in formula $Context, LowerLevelGoal \vdash Strategy$, the Strategy has become indicative, while the LowerLevelGoal is optative. These observations are coherent with the fact that proceeding from the business goal level to the lowest-level operational goals involves a sequence of decisions. Our description method is suitable for representing the progress of the decisional process, as well as the cause-effect relationships linking goals and strategies at the different levels.

V. Selecting a Strategy

Different strategies are characterized by different costs, effectiveness, risks, and benefits, so that choosing a strategy (i.e., exercising the option) implies that multiple characteristics of multiple strategies may need to be assessed. Therefore, in addition to the Goal, a Figure of Merit (FM) exists, whose value depends on the Context and the Strategy. The FM can be used in two ways. First, a constraint can be set on the FM. For instance, if cost is the FM, we can consider acceptable only strategies whose cost is below a specified cost threshold. Second, the FM can be used to comparatively assess different strategies, based on a Preference Criterion (PC) that ranks alternatives based on their corresponding values of FM. The PC may be a straightforward one when the FM is a single-objective one. However, FMs are often multiple-objective: for instance, a double-objective FM may address effort and development time. The application of the PC results in general in a partially ordered set of strategies, as some strategies may be deemed equivalent as for their FMs.

Making the FM and PC explicit shows that the selection of a strategy is not based only on the Goal; instead, it involves the optimization of characteristics that do not necessarily appear in the Goal. For instance, take the business Goal in the example, which should be interpreted as "Increase customer satisfaction by at least 20%." This Goal sets a constraint on the set of possible strategies used to reach it, but by no means does it explicitly indicate how to choose among competing strategies that satisfy it. In principle, one could choose any Strategy that satisfies the Business Level Goal in the given Context, regardless of the cost. However, in practice, the Strategy that minimizes the cost is likely to be preferred over the others.

Also, making the FM and the PC explicit provides guidance in the building of effective strategies, when no previously used strategies are available, or in the tailoring of existing ones or when there is a significant level of uncertainty, which is always present when making decisions. If so, we may not be able to identify the optimal Strategy with certainty, but the FM and the PC will help us at least reduce the set of strategies.

Summarizing, the FM and the PC need to be made explicit so that all ambiguities are removed as to why a specific Strategy is selected. Also, the analysis of the results obtained in the field will allow us to refine our decision processes.

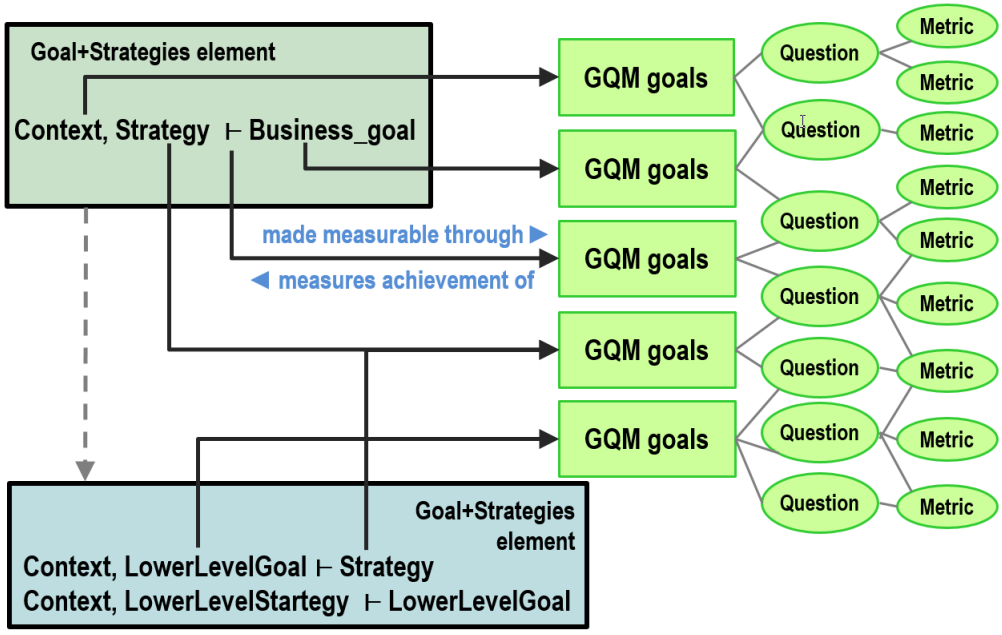

Fig. 2. Our proposal.

VI. What should be measured (and how)

Basili et al. provide the following indications for measurement [5]:

> *Associated with each GQM + Strategies element is a measurement plan that uses the GQM measurement and evaluation framework to specify how to evaluate the goal, what data to collect, and how to interpret that data. The nodes of each GQM graph consist of a measurement goal, which describes what knowledge needs to be gained from the measurement activity; a set of questions to be answered; the metrics and data items required to answer the questions; and an interpretation model that specifies how the data items are to be combined and what the criteria are for determining the goal's success.*

With respect to GQM + Strategies, Formula (1) provides clearer indications on what should be measured. While a single GQM plan is connected to a Goal+Strategy element [5], it is more natural and effective to associate specific measurement plans to each part of the entailment:

- Context: if the context description contains assumptions, it is generally a good practice to measure to what extent the assumptions are true.
- Goal: of course, we want to know to what extent the goal has been achieved. To this end, a GQM plan is typically attached to the business goal.
- Strategy: like the goal, we want to know to what extent the strategy has been applied. So, a specific GQM plan is typically defined for the strategy.

For sure, we want to measure the FM associated to a given entailment. In some cases, we could even have several FMs, each one representing a specific point of view. For instance, we could have a FM for top management and another one for the project manager. Measuring the FM usually requires measuring the elements the Context and the Strategy to which a FM refers. However, it must be noted that very often a FM concerns properties (e.g., the amount of resources used to implement a Strategy, or the time taken to complete the activities involved in a given Strategy) that belong to a sort of meta-level, and are possibly not considered in the "basic" measurement of Strategy. The quantification of Strategy selection criteria usually does not call for additional measures; instead, it is just a function of the computed FM.

As an example, let us consider the entailment *Plans based on reliable estimates of resource needs lead to more effective usage of resources, Resource allocation planning is improved* ⊢ *Available resources used more effectively*. Evaluating the FM involves measuring properties like the cost of planning, the increase of competence needed to perform better planning, the cost and the learning curve of tools used for planning, etc.

The entailment is usually assumed to be true. In other words, it is believed that the devised Strategy, correctly applied in the given Context, causes the full achievement of the Goal. However, it may happen that the Goal does not follow from the Context and Strategy. Measuring (i.e., looking for quantitative evidence of) this fact is therefore advisable. This usually involves verifying the connection between properties of the processes and products addressed by the Strategy and processes and products considered in the Goal. For instance, in the example's top level Goal and Strategy, one of the conditions that make the entailment true is that the cost of development depends on the usage of resources: this is usually true but not always so (e.g., when free resources are used). The interpretation model mentioned in [5] is clearly of great

importance, since the whole interpretation of the collected data depends on it. Nevertheless, in [5] it is not specified how the interpretation model should be defined; instead, it is delegated as part of the GQM plan. This is not advisable, in that the GQM itself is generally more oriented to refining goals into metrics than in prescribing how the collected data would be interpreted.

With our approach, the interpretations are generally made apparent by the formulae. Moreover, we do not have multiple GQM plans and graphs, as in GQM+Strategies; instead we have a single plan, with clearly interconnected elements, as shown in Figure 2 (which schematically represents a portion of the requirements hierarchy).

The connections between a strategy and its lower level goals are not emphasized by Basili et al., while they are clearly represented and measured in Figure 2.

VII. Related Work

The weakness of GQM in describing the software product or process that is the object of measurement were overcome by coupling GQM-compliant measurement tools with tools for modeling the product and process [12]. The work described here can be seen as a continuation of that work, in that here we provide the basis for coupling reasoning on business goals, user requirements, software development and –finally– measurement.

The need for linking business processes and Goal/Question/Metric paradigms has been felt since 2004 [13]. In [13], the authors define a measurement framework to support process analysts in assessing business processes by means of the GQM paradigm, to find useful indications about process performance, critical elements, change impact, and expected improvement. In our approach, the focus moves from a way to assess the quantitative and qualitative aspects of a business process to a way to precisely (possibly formally) describe business processes in a manner that is compliant with the GQM paradigm. The precise description of the business world and of company goals eases both the measurement of process aspects and the evaluation—both quantitative and qualitative—of the business and technical aspects of the process.

GQM+Strategies has been introduced for the first time in [4] to extend the GQM approach with the capability to create measurement programs that ensure a link between business goals and strategies, software goals, and measurement goals. The approach has been supported by the SAS tool to improve the definition of the context, assumption, and strategies [14]. In our paper, we adopt the extensions proposed in [4] to go further in the direction of representing the BW processes that are to be connected with GQM+Strategies. Our approach makes the representation of relevant relationships explicit, independently from the GQM.

In [15], the GQM+Strategies approach is adopted to perform business value analysis and to identify success/critical business goals. The paper clearly states that the various aspects of business value expressed and defined by goals require the knowledge and experience of the stakeholders to identify what elements (context, assumptions, strategies, goals) are valuable and appropriate for the company's success. In our paper, we aim at improving the process of describing the BW, in terms of the business domain, characteristics, goals, strategies and relationships with the software activities.

In [16], the author notes that the business level should be mapped into a Conceptual/Strategic level to clearly define the scope of the Business level in a generic way (i.e., outside the boundary of the software domain): the conceptual level is actually the highest organizational abstraction where an organization determines how to succeed in those activities that are strategic for the existence of the organization itself. This kind of mapping is quite easy with our approach.

VIII. Conclusions and Future Work

In this paper, we introduce a proposal to help organizations better represent their business goals and how to achieve them (mostly via software), and to link the business-oriented descriptions with measurement goals and plans. Our proposal is based on using Jackson's ideas on domain representation and uses concepts from GQM+Strategies, and allows for the precise description of the business domain, the business goals, the strategies, and their relationships with the software activities carried out supporting the strategies, and how strategies are selected. Thus, the most promising approach does not appear to consist in inventing a brand new technique or notation, but in leveraging on two existing techniques to make their joint use applicable in practice. The proposal also makes it possible to clearly and explicitly describe and therefore record the rationale behind the selection of strategies. A Figure of Merit of practical interest needs to exist, in addition to a Goal, for the evaluation of strategies in a given Context. A Preference Criterion must be defined so the different strategies can be ranked according to the values of their Figure of Merit.

A significant amount of future work remains to be done, including:

- Applying the approach to a set of real life business cases;
- Defining a fully coherent approach that enriches the GQM+Strategies methodology;
- Developing supporting tools to be integrated with existing GQM tools.

Acknowledgment

The work reported here was partly supported by Fondo per la Ricerca di Ateneo, Università degli Studi dell'Insubria.

References

[1] V. R. Basili and H. D. Rombach, "The tame project: Towards improvement-oriented software environments," IEEE Transactions on software engineering, vol. 14, no. 6, 1988, pp. 758–773.

[2] V. R. Basili and D. M. Weiss, "A methodology for collecting valid software engineering data," IEEE Transactions on software engineering, no. 6, 1984, pp. 728–738.

[3] G. Caldiera, V. R. Basili, and H. D. Rombach, "Goal question metric paradigm," Encyclopedia of software engineering, Volume 1, 1994, pp. 528–532.

[4] V. Basili, J. Heidrich, M. Lindvall, J. Munch, M. Regardie, and A. Trendowicz, "GQM+ strategies–aligning business strategies with software measurement," in First international symposium on empirical software engineering and measurement (ESEM 2007). IEEE, 2007, pp. 488–490.
[5] V. R. Basili, M. Lindvall, M. Regardie, C. Seaman, J. Heidrich, J. Münch, D. Rombach, and A. Trendowicz, "Linking software development and business strategy through measurement," Computer, vol. 43, no. 4, 2010, pp. 57–65.
[6] M. Jackson and P. Zave, "Deriving specifications from requirements: an example," in Proceedings of the 17th international conference on Software engineering, 1995, pp. 15–24.
[7] M. Jackson, "The meaning of requirements," Annals of Software Engineering, vol. 3, no. 1, 1997, pp. 5–21.
[8] ——, Problem Frames: Analyzing and structuring software development problems. Addison-Wesley Longman Publishing Co., Inc., 2000.
[9] V. Basili, J. Heidrich, M. Lindvall, J. Münch, C. Seaman, M. Regardie, and A. Trendowicz, "Determining the impact of business strategies using principles from goal-oriented measurement," in 9 Internationale Tagung Wirtschaftsinformatik, Wien, Austria, 2009, 2009, pp. 1–10.
[10] G. Caldiera, V. R. Basili, and H. D. Rombach, "The experience factory," Encyclopedia of Software Eng, vol. 1, 1994, pp. 469–476.
[11] L. Lavazza, "Business goals, user needs, and requirements: A problem frame-based view," Expert Systems, vol. 30, no. 3, 2013, pp. 215–232.
[12] L. Lavazza and G. Barresi, "Automated support for process-aware definition and execution of measurement plans," in Proceedings of the 27th international conference on Software engineering, 2005, pp. 234–243.
[13] L. Aversano, T. Bodhuin, G. Canfora, and M. Tortorella, "A framework for measuring business processes based on gqm," in 37th Annual Hawaii International Conference on System Sciences, 2004. Proceedings of the. IEEE, 2004, pp. 10–pp.
[14] V. Mandić and M. Oivo, "SAS: A tool for the GQM+ strategies grid derivation process," in Product-Focused Software Process Improvement: 11th International Conference, PROFES 2010, Limerick, Ireland, June 21-23, 2010. Proceedings 11. Springer, 2010, pp. 291–305.
[15] V. Mandić, V. Basili, L. Harjumaa, M. Oivo, and J. Markkula, "Utilizing GQM+ Strategies for business value analysis: An approach for evaluating business goals," in Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, 2010, pp. 1–10.
[16] S. A. Sarcia', "Is GQM+ Strategies really applicable as is to non-software development domains?" in Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, 2010, pp. 1–4.