

PREDICTIVE SOFTWARE ENGINEERING: DELIVERING EFFECTIVE BUSINESS SOLUTIONS THROUGH CUSTOM SOFTWARE DEVELOPMENT

Boris Kontsevoi

Intetics Inc., Naples, FL, USA

ABSTRACT

This paper explores the seven core principles of the Predictive Software Engineering (PSE) framework. These principles are designed to empower custom software development companies to deliver transparent and reliable solutions, all while adhering to predetermined budgets. The paper delves into each of the seven principles: Meaningful Customer Care, Transparent End-to-End Control, Proven Productivity, Efficient Distributed Teams, Disciplined Agile Delivery Process, Measurable Quality Management and Technical Debt Reduction, and Sound Human Development.

KEYWORDS

Agile, Disciplined Agile Delivery, Distributed Team, Predictive Software Engineering, Measurable Quality Management and Technical Debt Reduction System (MQM&TDR).

1. INTRODUCTION

Traditional software development often suffers from unpredictable elements and subjective decision-making. Predictive Software Engineering (PSE) addresses this issue by introducing a set of seven principles, informed by over 29 years of experience. These principles guide development teams towards a more standardized, controlled, and transparent approach, transforming software creation into a precise and predictable engineering endeavour.

2. MEANINGFUL CUSTOMER CARE

Customer-centricity remains the cornerstone of success for any business, regardless of its market orientation (B2B, B2C, etc.). Customer satisfaction, brand loyalty, and ultimately, customer acquisition volume collectively define an organization's success. When an organization's offerings fail to resonate with target audiences and translate into commercial success, it signifies that it has not yet maximized its potential.

Therefore, to objectively assess progress towards organizational goals, measuring customer satisfaction and feedback through key performance indicators (KPIs) becomes paramount. Should these metrics reveal an unfavourable trajectory, the focus should promptly shift towards transforming dissatisfied customers into ardent brand advocates. This objective can be achieved through the implementation of eight key strategies:

1. Streamline the onboarding process.
2. Offer a dedicated customer portal for easy access and engagement.
3. Establish a clear governance model for efficient issue resolution.
4. Define a transparent escalation path for handling complex concerns.
5. Implement early problem discovery methods to proactively address issues.
6. Conduct comprehensive analysis of customer complaints to identify root causes.
7. Proactively manage customer satisfaction through consistent monitoring and engagement.
8. Set realistic expectations to avoid disappointment and build trust.

3. TRANSPARENT END-TO-END CONTROL

Transparency, characterized by openness, trust, and cooperation, remains a scarce commodity in the contemporary business landscape. Despite its relative rarity, achieving transparency offers significant advantages. Stakeholders, particularly clients, perceive transparent organizations as less risk-prone and more reliable entities, fostering a collaborative environment conducive to long-term success. This paper explores the multifaceted approach required to cultivate transparency within organizations, both internally and externally.

External Transparency:

- **Governance and Project Meetings:** Regularly convening governance and project status meetings with external stakeholders serves as a key mechanism for maintaining information dissemination and fostering trust. These forums provide a platform for open communication, allowing clients to actively engage with the organization and address any concerns promptly.
- **Agile Methodologies:** The implementation of Agile methodologies plays a crucial role in promoting real-time visibility into project progress. This iterative approach enables stakeholders to remain informed and engaged throughout the development process, fostering adaptability and collaborative problem-solving.
- **Collaboration Tools and Project Portals:** The effective utilization of technology-driven solutions such as teamwork collaboration tools and project portals empowers clients with real-time access to project details. This fosters a sense of ownership and transparency, enhancing stakeholder confidence in the organization's commitment to open communication.
- **Process Audits and Proactive Monitoring:** Implementing regular process audits allows organizations to proactively identify and address potential roadblocks. Additionally, shifting from reactive monitoring to a proactive approach ensures issues are identified and addressed before they escalate, demonstrating a commitment to continuous improvement and mitigating potential risks for stakeholders.
- **On-Site Visits:** Scheduling on-site visits at key project milestones serves as a powerful tool for strengthening open communication with clients. These visits demonstrate the organization's commitment to transparency and collaboration, fostering trust and building stronger relationships with external stakeholders.

Internal Transparency:

- **Leadership Engagement:** Fostering a culture of transparency necessitates active and consistent engagement from the organization's leadership. Conducting regular meetings with C-level executives facilitates clear communication and alignment across all organizational levels, ensuring all stakeholders are informed and aligned with company goals.

- **Departmental Check-ins:** Establishing regular departmental check-in meetings with directors fosters information flow and collaboration across various divisions. This practice promotes transparency by breaking down silos and ensuring all departments are working towards common objectives.
- **Project Management Engagement:** Regular check-ins with project managers serve a two-fold purpose. Firstly, they ensure that project progress is closely monitored, and potential issues are addressed swiftly. Secondly, these interactions foster a transparent environment where project managers feel comfortable voicing concerns and seeking guidance from leadership.
- **Talent Management Integration:** Including the talent management team in fostering collaborative work environments underlines the organization's commitment to internal transparency. This collaborative approach empowers employees to voice concerns and actively participate in decision-making processes, promoting a sense of ownership and trust within the organization.
- **Team Building and Company Events:** Investing in team-building events and company gatherings strengthens internal communication and interpersonal relationships within the organization. These events provide informal spaces for information exchange and collaboration, fostering a sense of trust and transparency among employees.

Cultivating a culture of transparency requires a comprehensive and multifaceted approach. By implementing the strategies outlined above, both internally and externally, organizations can establish themselves as reliable and trustworthy partners. This commitment to openness fosters trust with stakeholders, facilitates collaboration, and ultimately positions the organization for long-term success in today's competitive business environment.

4. PROVEN PRODUCTIVITY

Evaluating software engineer productivity remains a persistent challenge across diverse organizations. This paper explores two prominent tools employed to gauge productivity and velocity during development:

- Measurable Quality Management and Technical Debt Reduction System (MQM&TDR) (addressed in a separate section)
- Software Development Efficiency Key Performance Indicator (KPI)

The Software Development Efficiency KPI encompasses ten sub-KPIs designed to assess various development aspects:

- **Static code analysis (project & individual):** Evaluates overall and individual developer code quality through automated tools.
- **Planned vs. actual work:** Compares planned workload with actual completion, providing insights into project planning efficiency.
- **Bugs per line of code (LOC):** Measures defect frequency relative to code volume, indicating code quality and potential rework effort.
- **Opened bugs per feature:** Assesses defect prevalence within individual features, aiding in identifying areas for improvement.
- **Templates/algorithms/standard libraries missed:** Evaluates adherence to established practices and best practices utilization.
- **Missed exception/error handling:** Assesses implementation of robust error handling mechanisms, indicating code robustness and user experience.

- **Security problems:** Identifies potential security vulnerabilities within the code, highlighting critical areas requiring immediate attention.

To ensure effectiveness and integrity of the Software Development Efficiency KPI, several key guidelines are recommended:

- **Double code review:** Two developers independently review the same code for accuracy and comprehensiveness.
- **Documented code review:** Meticulously document the entire code review process for auditability and future reference.
- **Management review and approval:** Team managers or leaders should comprehensively review and approve the KPI results to ensure validity and alignment with organizational goals.
- **Anti-cheating tools:** Utilize tools that mitigate manipulation and simplify the review process.
- **Code review volume limitation:** Review smaller code segments (ideally less than 500 lines) to enhance focus and thoroughness during assessment.

5. EFFICIENT DISTRIBUTED TEAMS

The dissolution of geographical constraints has enabled organizations to embrace distributed teams, fostering the creation of a geographically diverse workforce. This model offers several advantages, including:

- **Time Effectiveness:** Distributed teams, spanning various time zones, ensure continuous coverage. By strategically placing development centers, organizations can guarantee at least one team is always operational, particularly valuable for entities requiring 24/7 support.
- **Cost Savings:** Hiring from diverse locations allows organizations to potentially reduce costs associated with salaries, rentals, and operating fees.
- **Talent Pool Diversification:** Distributed teams grant access to a virtually limitless pool of highly skilled talent, broadening the candidate pool beyond geographical limitations.

However, physical distance can introduce challenges such as communication issues and decreased efficiency. To ensure optimal performance within distributed teams, adhering to specific guidelines is crucial:

- **Task Design:** Tasks should be meticulously designed to facilitate collaboration, considering factors such as time zone differences, industry regulations, and individual workloads.
- **Role Clarity:** Roles and responsibilities of each team member should be clearly defined and communicated to avoid confusion and overlap.
- **Effective Communication:** Utilizing established and well-defined communication strategies, such as closed-loop communication, is paramount.
- **Loose Coupling:** Tasks should be loosely coupled, minimizing dependency on individual members' progress, and fostering overall team agility.
- **Mutual Performance Monitoring:** While maintaining individual responsibilities, team members can mutually monitor performance, fostering a sense of accountability and collaboration [1].

Beyond these considerations, social aspects also require attention. Team members from diverse backgrounds may exhibit differing communication styles and cultural norms. There are measures for addressing this:

- **Effective Leadership:** Providing strong and effective leadership is critical. Leaders should be adept at motivating team members, fostering cohesion, and offering necessary coaching.
- **Shared Experiences:** Cultivating common ground among team members is essential. When feasible, conducting shared training sessions in a centralized location can further enhance team unity.

By recognizing the benefits and adhering to these best practices, organizations can effectively leverage distributed teams to achieve their business objectives and harness the advantages of a geographically diverse workforce.

6. DISCIPLINED AGILE DELIVERY PROCESS

While Agile methodologies remain the predominant framework for software development, a vast array of adaptations exist. This paper posits that Disciplined Agile Delivery (DAD) emerges as a particularly advantageous variant for service organizations.

Unlike traditional Agile approaches, DAD employs a distinct lifecycle structured into three primary phases: inception, construction, and transition [2].

A. Inception Phase:

- **Vision Definition:** The project vision is meticulously defined, establishing a clear roadmap for development efforts.
- **Stakeholder Alignment:** Collaboration with stakeholders is crucial, ensuring project vision aligns with their expectations and objectives.
- **Initial Planning:** This phase encompasses crafting the initial project plan, outlining the preliminary technical strategy, and identifying core requirements.
- **Team Formation:** The initial project team is assembled, bringing together the necessary expertise and resources.

B. Construction Phase:

- **Adapting to Change:** Continuous adaptation to evolving stakeholder needs remains a central focus during this phase.
- **Iterative Development:** The focus shifts towards delivering a potentially usable solution through iterative development cycles.
- **Deployment Readiness:** Efforts are directed towards achieving a deployable release state.
- **Quality Assurance:** Maintaining or enhancing existing quality levels remains paramount throughout the construction phase.
- **Risk Mitigation:** Addressing the most significant project risks becomes a key priority during this phase.

C. Transition Phase:

- **Production Readiness:** This phase ensures the developed solution is fully prepared for deployment into the production environment.
- **Stakeholder Preparation:** Equipping stakeholders with the necessary knowledge and resources to effectively receive the solution is crucial.
- **Deployment Execution:** The final stage involves the successful deployment of the solution into production, marking the culmination of the project lifecycle.

By embracing the distinct phases and tailored approach of DAD, service organizations can leverage the agility and adaptability inherent in Agile methodologies while addressing the specific needs and challenges they encounter within their domain.

7. MEASURABLE QUALITY MANAGEMENT AND TECHNICAL DEBT REDUCTION

The Measurable Quality Management and Technical Debt Reduction System (MQM&TDR) emerges as a comprehensive framework designed to assess various facets of software development projects. Its core functionalities encompass:

- **Technical Debt Measurement:** MQM&TDR facilitates the quantification of technical debt, enabling proactive management and mitigation strategies to prevent accumulation and potential future issues.
- **Software Quality Assessment:** The system provides an unbiased evaluation of software quality across various dimensions, encompassing source code quality, usability, security, performance, and more.
- **Business Efficiency Determination:** MQM&TDR aids in determining the product's business efficiency, offering valuable insights for stakeholders.

The benefits of MQM&TDR extend beyond technical teams, offering advantages to various stakeholders:

- **Testers and Developers:** Gain objective feedback on their work, enabling continuous improvement and quality enhancement.
- **Managers:** Receive insights into the project's overall capabilities, facilitating informed decision-making.
- **Users:** Benefit from a reliable and well-performing final product.
- **Investors:** Utilize MQM&TDR's findings to assess investment risks and determine fair market value with greater accuracy.

The system's evaluation encompasses a comprehensive spectrum of software components:

1. Source Code Quality
2. Usability
3. Security
4. Performance
5. Business Logic
6. Solution Architecture
7. Data Quality
8. Open-Source Code Use

By leveraging MQM&TDR, various stakeholders can benefit from the following:

- **Proactive Technical Debt Management:** MQM&TDR enables the identification and mitigation of technical debt before it becomes a significant burden.
- **Reduced Costs:** Redevelopment and support costs can be significantly minimized through proactive quality management.
- **Informed Progress Evaluation:** MQM&TDR facilitates a data-driven approach to progress evaluation, fostering informed decision-making.
- **Predicted Business Efficiency:** The system provides insights into the product's anticipated business efficiency, allowing for strategic planning.
- **Comprehensive Quality Analysis:** MQM&TDR offers a holistic assessment of software quality, encompassing various critical aspects.
- **Detailed Feature Analysis:** The system provides in-depth analysis of key product features, aiding in refinement and optimization.
- **Compliance Verification:** MQM&TDR helps ensure adherence to relevant standards and regulations.
- **Actionable Improvement Recommendations:** The system offers specific recommendations for improvement, guiding ongoing development efforts.

Given its multifaceted functionalities and demonstrably positive impacts, MQM&TDR presents a compelling approach for enhancing software development projects across diverse organizations.

8. SOUND HUMAN DEVELOPMENT

While the language of human resources often frames individuals as mere resources, a more nuanced perspective recognizes employees as autonomous agents. These agents possess the capacity for independent decision-making, skill development, and goal setting. By prioritizing human development, organizations can cultivate not only a more skilled workforce but also a more fulfilled and loyal employee base [3].

This paper explores several key strategies for promoting human development within organizations:

A. Benefits: Expanding beyond standard benefits (e.g., medical insurance, compensation, office amenities) to incorporate training and certification programs alongside language learning opportunities (e.g., English courses) demonstrates a commitment to employee development.

B. Mentorship Programs: Implementing internship and mentorship programs offers numerous benefits. Interns gain invaluable practical experience working on real-world projects, transcending purely theoretical knowledge. This exposure extends beyond technical skills in programming and testing, encompassing communication and process understanding. Moreover, successful mentorship programs can often lead to retention, with interns transitioning into permanent roles after the program concludes, fostering a sense of team continuity and internal knowledge transfer.

C. Performance Review and Career Growth: Establishing individual development plans (IDPs) for each employee is crucial. Through collaborative sessions involving the employee, talent management personnel, and relevant team members, organizations can gain insights into employee aspirations and career goals. Leveraging these insights, career paths can be charted, outlining potential developmental opportunities and advancement possibilities. Furthermore, implementing regular performance reviews using a combined approach is recommended. This approach involves self-assessment by employees followed by managerial feedback, culminating

in a collaborative discussion to identify areas for improvement and establish actionable steps for the following review cycle.

D. Skills Improvement: Organizations must demonstrate a continuous commitment to facilitating skill development among their employees. This commitment can be manifested through three key strategies:

- **Engaging external experts:** Seeking external guidance from domain specialists allows employees to access cutting-edge knowledge and industry best practices.
- **Providing internal and online training:** Offering a diverse range of internal training programs and e-learning resources empowers employees to pursue self-directed learning and upskilling opportunities.
- **Establishing Centers of Excellence (CoEs):** Creating dedicated CoEs within the organization fosters knowledge sharing and expertise concentration, allowing employees to learn from industry leaders and internal specialists within specific domains.

By implementing these comprehensive strategies, organizations can move beyond the limitations of traditional human resource management and embrace a human-centric approach that prioritizes individual development, career advancement, and overall employee well-being. This shift in perspective fosters a more engaged, motivated, and skilled workforce, ultimately contributing to organizational success and sustainability.

9. RISKS COVERED WITH PSE

By embracing Predictive Software Engineering (PSE) and adhering to the outlined principles, software development teams can effectively mitigate a range of potential risks, including:

- **Executive Disengagement:** The principles of Transparent Control and Meaningful Care foster open communication and stakeholder buy-in, minimizing the risk of insufficient executive support.
- **Scope Creep:** Transparent Control promotes clear and well-defined project scope, reducing the likelihood of ambiguity and uncontrolled expansion.
- **Inaccurate Estimates:** Disciplined Agile methodologies encourage iterative development and continuous estimation refinement, mitigating the risk of inaccurate initial estimates.
- **Resource Constraints:** Distributed Teams empower organizations to leverage a wider talent pool, potentially addressing resource limitations.
- **Skill Gaps:** Sound Human Development practices, like training programs and skill-building initiatives, equip teams with the necessary skills and knowledge, minimizing the potential for inadequate training.
- **Unclear Decision-Making:** Meaningful Care emphasizes open communication and collaborative decision-making, reducing the risk of ambiguity and confusion arising from unclear decisions.
- **Infrastructure Delays:** Transparent Control promotes proactive planning and communication, allowing for the timely identification and mitigation of potential infrastructure-related delays.

Through the adoption of PSE and its associated principles, software development teams can navigate these common challenges and increase the likelihood of project success.

CONCLUSION

Custom software development often encounters bottlenecks hindering efficient and reliable delivery. Predictive Software Engineering (PSE) emerges as a novel framework designed to address these challenges and reconstruct a more robust approach to software development service delivery.

PSE seeks to mitigate the inherent uncertainties often associated with the creative aspects of programming. By establishing a structured and predictable framework, PSE aims to elevate software engineering to its intended state: a precise and controlled discipline.

Comprised of seven distinct yet interconnected concepts, the PSE framework fosters transparency, controllability, and inherent predictability within the software development process.

ACKNOWLEDGEMENTS

While proprietary to Intetics and drawing upon their extensive experience in custom development, the framework's developers emphasize its broader industry value.

Recognizing the limitations of a solely individual effort, Intetics encourages collaborative industry-wide engagement in further refining and expanding upon the PSE framework. They emphasize that the framework's objective transcends individual company success, aiming to elevate the entire software development profession. This call for collaboration underscores the shared benefit of collective knowledge sharing and continuous improvement, prioritizing the advancement of the field above competitive advantage.

It is crucial to acknowledge that PSE remains an ongoing project, continuously evolving through research and development. The initial research, led by Boris Kontsevoi and Sergei Kizyan, laid the groundwork for this promising framework, building upon established global engineering practices. By promoting collaboration and knowledge exchange, the development of PSE can be accelerated, ultimately contributing to a more efficient, reliable, and predictable software development landscape.

REFERENCES

- [1] Larsson, A., Törlind, P., Karlsson, L., Mabogunje, A., Leifer, L., Larsson, T., Elfström, B-O.: Distributed team innovation – a framework for distributed product development. In: 14TH INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN 2003, ICED'03, vol 322, pp. 1-10. Design Society, Stockholm (2003). ISBN: 1-904670-00-8.
- [2] Ambler, S. W., Lines, M.: Introduction to Disciplined Agile Delivery. Crosstalk Journal 40, 7-11 (2013).
- [3] United Nations Development Programme Human Development Report 2016, <https://www.undp.org/publications/human-development-report-2016>, last accessed 2018/10/10.
- [4] Kontsevoi, B., Terekhov, S.: TETRA™ Techniques to Assess and Manage the Software Technical Debt. *Advances in Science, Technology and Engineering Systems Journal* 6 (5), 303-309 (2021). DOI: <http://dx.doi.org/10.25046/aj060534>.