Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 159 (2019) 1777–1786

Procedia Computer Science

www.elsevier.com/locate/procedia

23rd International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, KES 2019, 4-6 September 2019, Budapest, Hungary

IT Crisisology: The New Discipline for Managing Software Development in Crisis

Dr. Sergey V. Zykov [a,b,c,] *

[a] *National Research University Higher School of Economics, 20 Myasnitskaya St., Moscow, 101000, Russian Federation*
[b] *MEPhI National Research Nuclear University, 31 Kashirskoe Shosse, 115409 Moscow, Russian Federation*
[c] *I.M. Sechenov's First Moscow State Medical University, 8-2 Trubetskaya St., Moscow 119991, Russian Federation*

Abstract

This paper discusses crises in software product development. It defines a development crisis and presents the new discipline of IT Crisisology, which systematically studies and addresses crises in software production. Essential ingredients of this IT Crisisology discipline include models, methods, tools, patterns and practices. The IT Crisisology framework developed addresses technology, business and human factors; for each kind of factors, it uses a dedicated set of the above-mentioned ingredients. These include tradeoff optimization, agility matrix, models for knowledge transfer and data lifecycle, to name a few. For enterprise scale development, these ingredients are further enhanced and improved. In case of systematic application, the suggested IT Crisisology framework allows for better disciplined, predictable, and manageable software product development, even in the event of a crisis.

Keywords: IT crisisology; software product development; lifecycle management; agile methodology

* Corresponding author. Tel.: +0-000-000-0000 ; fax: +0-000-000-0000 .
E-mail address: szykov@hse.ru

10.1016/j.procs.2019.09.349

1. Introduction

Crisis is an imbalance between the expectations of clients and the exact behaviour of the product. It was due to rapid increases in computer power and the complexity of unhandled problems. The term “crisis” is used in computer science to describe difficulties of writing useful and efficient computer programs in a scheduled time. Let us define crisis in software production as a situation of either premature project termination or insufficient quality/late product delivery due to imbalance of project resources & product constraints.

Let us define IT Crisisology as a discipline that studies crises in software production, including their monitoring, forecasting, detection, mitigation, resolution, resilient and adaptive (either proactive or reactive) responding, and prevention. Crises affect both software project and software product. In software projects, crises occur in the form of over budgeting, late delivery of projects, unmanageable projects whereas in software products they manifest themselves as the products being inefficient, low quality, essentially different from customer requirements, difficult to maintain, and even undelivered.

Nomenclature

A	radius of
B	position of
C	further nomenclature continues down the page inside the text box

- Prior Knowledge
- Continuous improvement: feedback from the system, the customer, and team members throughout the project. The aim is to improve future iterations using past records.
- Customers are informed and work closely with the team. Customers see the deliverables, share the input, and have an impact on the end product. The aim is to meet customer requirements.
- Changes: it is easy to accept and set changes with a shorter planning cycle. The aim is to refine and reprioritize items.
- Fast and high-quality project delivery: the product is broken down into manageable units. Team focus is high-quality development and testing. Bugs and errors identified and solved quickly; the product is delivered fast, with a reasonable schedule and budget.
- Team interaction: frequent and face-to-face communication results in high quality product.
- Continuous goal clarification: with a clear goal, development adapts to evolving requirements as project progresses within scope
- Metacognition

2. Three “Pillars” of IT Crisisology

The first “pillar” refers to technical requirements; these include quality attributes (portability, security, performance) and system-level framework such as integrated development environment (IDE), database management system (DBMS), and programming language (PL). The second “pillar” refers to business constrains; this occurs in the form of project delimiters (i.e. budgets, time to market) as well as the problem domain in which this crisis occurs. The third “pillar” is known as the human factor; this encapsulates such communication attributes as teambuilding ability, negotiation skills and the ability to transfer knowledge to other people. Let us refer to these “pillars” as the technology, business and human factors, i.e. “T”-factor, “B”-factor and “H”-factor. The idea is that removing any of the above “pillars” results in a “global” crisis (as it would with removing any leg of a three-legged stool), whereas neglecting any of these results in a “local” crisis due to their imbalance.

3. Myths about Crisis

In April 12, 1959, the US President J.F.Kennedy said that a “crisis” meant an “opportunity” [20]; however, the Chinese word for crisis is typically composed of these two characters: “danger” and “situation” [21]. Notably, at this point of time the “Crisisology” was born as a phenomenon. As the years went by, researchers argued whether the crisis in Software Engineering is over or it still exists. In 2008, Coliburn et al., argued to prove that there was no software engineering crisis [2]. However, in 2006, Buettner et al., argued conversely, stating that software engineering was indeed in a crisis [3]. All these happened, because of the fundamental differences in the lifecycles of software and material products, and due to human factor dependencies.

4. IT Crisisology

This paper presents IT Crisisology, the discipline that systematically studies IT crises phenomena and their management. This discipline addresses business, technical and human factors; it applies a carefully selected blend of the software engineering models, methods and tools that result in manageable (i.e. measured and predictable) projects, better quality products, conquering product complexity, ambiguity, and managing project uncertainty.

IT product management differs from financial management and material production management in many ways, some of which are its product type, which is up to 100% non-material, and its lifecycle (i.e. implementation and maintenance cost). The other key reason is communication ambiguity, as the developer and the client sides typically speak in difference languages. This is evident even from the Google search results, which report 612M for the “Crisis management” query, 629M for “Financial crisology” and 1.2K for “IT crisisolgy”.

In 1968, the NATO organized the first conference in software engineering in Garmisch, Germany, in order to find a solution to what became known as “crisis” at that time. Among the key delegates of this conference were Friedrich Bauer, Alan Perils, Edger Djikstra and Peter Naur. However, software has been in crisis for over 50 years. A number of models were developed due to the crisis; one example is the triangle crisis triangle (Fig. 2) based on the “Iron” (project) triangle (Fig. 1). Let us mention the “Quality”, which, as a derivate of the “3D”, deals with adjusting tradeoff and managing the project budget, project scope and project schedule.

The “crisis triangle” consists of three zones. These include the “comfort” (i.e. green) zone, where there is little need for management since the actions proceed as planned, and this guarantees quality deliverables. The other zone is the “tradeoff” (i.e. yellow) zone, where a number of factors may be adjusted, and immediate actions are required in other to guarantee the quality. The third zone is the “crisis” (i.e. Red) zone; in this critical area, immediate actions are required to produce even a negotiable quality.

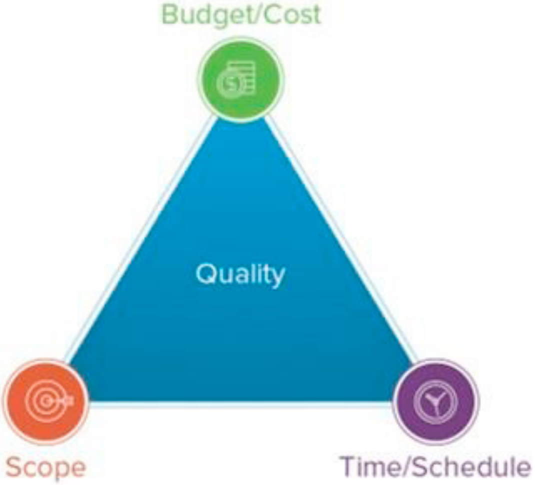


Fig.1. The “iron” (project) triangle

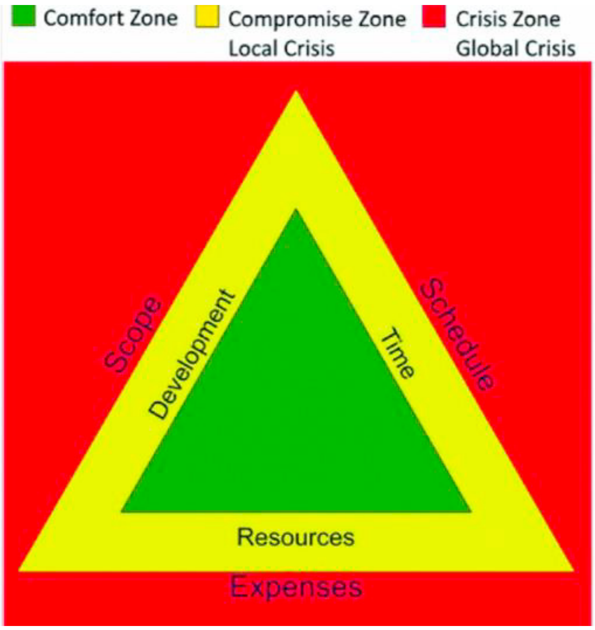


Fig. 2. The “crisis” triangle

Dealing with this crisis triangle, a paramount factor is the communication between the developer and the customer. Shannon information theory fits the “developer-and-customer” system, where the key issue is transferring messages without loss, especially in a “noisy” environment. In this model, a crisis typically results from an information loss, and in case of a large-scale system, such a loss can happen due to complexity. To compensate negative issue and to conquer the complexity, a feedback (either negative or positive) is mission-critical.

The soft skills that result from the so-called human factors (such as communication, negation, conflict management and teambuilding), are based on that is psychology or sociology and highly essential for the software engineers; however, these are initially rare for the technical staff (i.e. software developers). These human factors can be broken-down into “seven principles” of knowledge transfer:

- Prior Knowledge
- Knowledge Organization
- Motivation
- Practice + Feedback
- Mastery
- Climate
- Metacognition

Surprisingly, the human factors result in crises more often than the technical factors. In order to manage these, metrics to determine if the product development goals are met, how to achieve these goals and what is required, why the goals are necessary, time, priority and due date for each deliverable are established. These ingredients, taken together, are often referred to as SMART communication. Also, adaptive development practices such as teambuilding, Myers Briggs personality psychological types, Situation leadership model and Personal development model are considered [4].

This difference between developers and customers inspires the need for trade-off-based architecture (ACDM/ATAM). ACDM/ATAM reduces uncertainty and helps avoid crisis. How this model ensues that crisis is avoided by reducing uncertainly (fig. 3). ACDM defines the main stages of work on the coordination of design and rapid, adaptive development of high-quality architectural design. It helps teams to analyze, identify, and build architecture “drivers” in the early stages of software development. The architecture “drivers” help to design, evaluate and test a software development project. The results of the project evaluation contribute to the refinement of the architectural design. The ACDM method includes these three main components:

- Processes: these involve iterative development and clarification of the architectural project
- Project evaluation: this determines whether the project is ready for production and identifies problems important to architecture
- Architectural testing: this solves technical problems identified during the assessment, serves the basis for the refinement of the project, and requires (re)assessment until the product is ready for production

In ACDM, the main goal is software design. This is achieved by performing these tasks: identifying, analyzing and organizing architectural drivers, architectural project management and its documenting, project evaluation, management, iterative refinement of software design to reduce risk, assessment, planning and correction of the system/product at the level of architecture

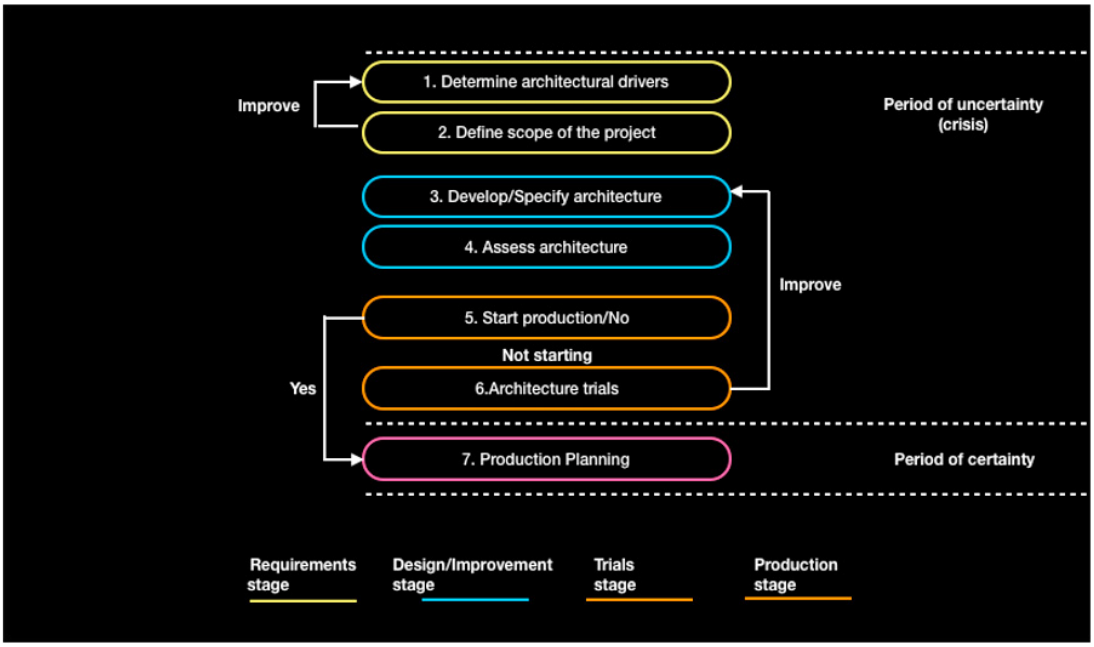


Fig.3. ACDM/ATAM: SW Development Lifecycle

5. Agile software development

This is a set of approaches based on iterative development, where requirements and solutions evolve through the collaborative effort of teams and their customers, or end-users. It provides a response, which is rapid and flexible to the change like the crisis.

The general Agile software development lifecycle includes such phases as: Plan, Requirements Analysis, Design, Development, Test and Deployment. A number of software development methodologies follow the Agile way. These include: Scrum, Extreme Programming (XP), Feature-driven Development (FDD), Adaptive Software Development (ASD), Lean Software Development (LSD), Kanban, Dynamic Software Development Method (DSDM), and Crystal Clear.

Agile methodologies help crisis management by employing the following techniques and practices:

- Continuous improvement: feedback from the system, the customer, and team members throughout the project. The aim is to improve future iterations using past records.

- Customers are informed and work closely with the team. Customers see the deliverables, share the input, and have an impact on the end product. The aim is to meet customer requirements.
- Changes: it is easy to accept and set changes with a shorter planning cycle. The aim is to refine and reprioritize items.
- Fast and high-quality project delivery: the product is broken down into manageable units. Team focus is high-quality development and testing. Bugs and errors identified and solved quickly; the product is delivered fast, with a reasonable schedule and budget.
- Team interaction: frequent and face-to-face communication results in high quality product.
- Continuous goal clarification: with a clear goal, development adapts to evolving requirements as project progresses within scope

6. Lifecycle management in the enterprises

In general, the lifecycle management in enterprises is similar to a person's life in a number of aspects. The organization is born in creative and entrepreneurial agony experiencing the difficulties and joys of adolescence. It reaches maturity, and then many organizations begin to grow old and decline as young and dynamic competitors gradually replace them. The old age is usually followed by organizational death.

However, unlike people's lives, the lifecycle of organization is not limited to a certain time period. Examples of organizations are known that have been in their blossoming and maturity for decades; this can be observed in companies like Google, Apple and Amazon.

7. Change management and consequences

Often, a problem or an opportunity created by a change, leads to a solution that will cause an even greater change, and as a result the customers face a new reality with a new set of problems or opportunities. When the systems change, they collapse and disintegrate. However, in order to break up and disintegrate, it is not necessary to be old. Just as young people sometimes commit suicide, young systems can also disintegrate. Therefore, regardless of the age of the system, the cause of its disintegration is change, and the faster the change, the faster the disintegration that manifests itself in what we call problems. The lifecycle of an enterprise proceeds in a number of stages, which are summarized below.

The Courtship phase precedes the emergence of an organization that has not yet come into being and exists only as an idea. During the Courtship, the emphasis is typically on the ideas and opportunities that the future promises. The potential founder of the company experiences a burst of enthusiasm and willingly tells everyone about how wonderful his or her idea is.

The Infancy phase of the development of the organization is not important to what someone does rather than he or she thinks. The question that the founders must answer and which they ask their employees, looks like this: "What did you do? Did you manage to sell something, produce it, or bring it to an end?" Infant companies face an amazing paradox. The higher their risks, the higher their loyalty should be in order to ensure the achievement of success. At this phase of courtship, the founders must be dreamers capable of developing devotion to their dreams. However, as soon as a company enters the period of Infancy, the risk increases and it begins to demand persistent, result-oriented founders who are no longer dreamers.

The "Let's Go" phase is a dramatic period of abrupt transition from love to hatred. Workers may not like their leaders, but they continue to fear and respect them. If a company falls into the trap of the founder, it means that when the founder dies, the company dies. In order for the company to be able to save the hard-earned wealth, it must move from management based on intuition and position (used in the "Come-Come" phase) to a more professional management. This process is carried out at the stage of the company's Youth. If the company does not implement such a transition, it falls into the founder's trap or into a family trap.

The Youth phase is characterized by a number of problems, such as: (i) a conflict between partners or decision-makers, (ii) a temporal loss of vision, (iii) unsystematic way of delegation of authority; therefore, the rules are established but not observed. The other problems that occur include: (i) inadequacy of goals, (ii) suspension of the founder, and (iii) rapid weakening of mutual trust and respect.

The Flourishing phase is for the organizations that have reached the stage of "Come-Come" and create new companies, i.e. the new business units that have their own products, as well as their own production and marketing capacities. These new units can exist for themselves. Like an adult tree, a company that has reached blossom also contains seeds of new companies in its fruits. They are not just new functions. They are new profit centres. Organization of the Flourishing period is typically a group of profit centres that share some functions for the sake of economies of scale, for the benefit of coordination or for the sake of preserving the critical mass for further creation. Blossoming, is an optimal state of the life cycle, achieving a balance between self-control and flexibility.

Symptoms of the Aging organization emerging from the state of prosperity can be seen in its financial statements. Financial statements also help to detect a disease when abnormal symptoms appear in them, and we can only hope that they saw the problems before they acquired the nature of the pathology. The purpose is to identify the symptoms of impairment when there is still the possibility of preventive treatment of the disease.

At the Aristocracy phase, organizations reduce expectations of growth, show little interest in conquering new markets, mastering new technologies and expanding borders, focus on past achievements rather than future vision; are suspicious of the changes; reward those who do what they are told to do; are more interested in maintaining interpersonal relationships; spend money on control systems and on the arrangement of premises; worry about how to do rather than what and why; require adherence to traditions, and compliance with formalities.

Companies that have reached the Salem City phase have the following characteristics. People focus on who created the problems, not on what to do to solve them. Problems are personified; instead of dealing with the problems of the organization, people are involved in the conflicts, accuse and discredit each other. The organization includes universal paranoia. In the undercover struggle, everyone is included; nobody has time to deal with the needs of the client.

At the phase of Bureaucracy, companies are unable to generate enough resources on their own. They justify their existence by the simple fact that they are of interest to another organization that is ready to provide their support. The organization needs an artificial life support system that allows delaying the Death. And what ensures the provision of such a system is a political decision.

Organizational Death phase typically happens when the lack of resources to reward employees of the organization for their work. The organization is dead when none of its members want to appear at work: there is no reason for this anymore. Death comes at a time when nobody has the commitment of the organization.

The four managerial roles to account for the above phases, are: Purposeful (P), Administrative (A), Entrepreneurial (E) and Integrative (I).

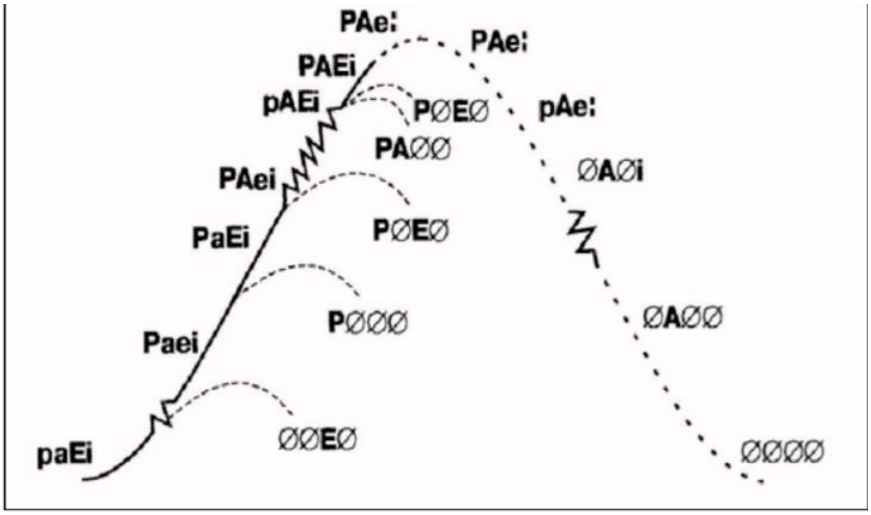


Fig. 4. The Role of Interaction

8. The Adizes Methodology

Adizes program teaches "leaders of change" when they need to focus their efforts on external integration when, on the inside, and when in both directions at once. The tools of this systematic approach support the processes of internal, external and internal and external integration. This therapeutic intervention allows analyzing the state of organizations and determining, depending on their position on the life-cycle curve, what is likely to happen (i.e. predicting crises) – see Fig. 4. The Adizes methodology outlines the therapy, depending on the stage of the life cycle, which the organization itself implements. The structure of each organization, its management style, remuneration system, planning process, objectives and other features may be desirable or destructive depending on the organization's position on the life-cycle curve.

9. Optimized Spiral Model

Every lifecycle stage of the software system development can be optimized. To optimize the lifecycle, i.e. to adapt it for crisis conditions, a complex methodology is required. This complex methodology is a combination of a set of models, methods, CASE tools and practices. The methodology is process-based, and it has six stages, each of which produces certain deliverables in terms of software product components and their connectors. At the most abstract level, these are key concepts of the product and certain relationships between these concepts. Next, high-level architectural modules and interfaces follow [19]. These stages are shown in Fig 5

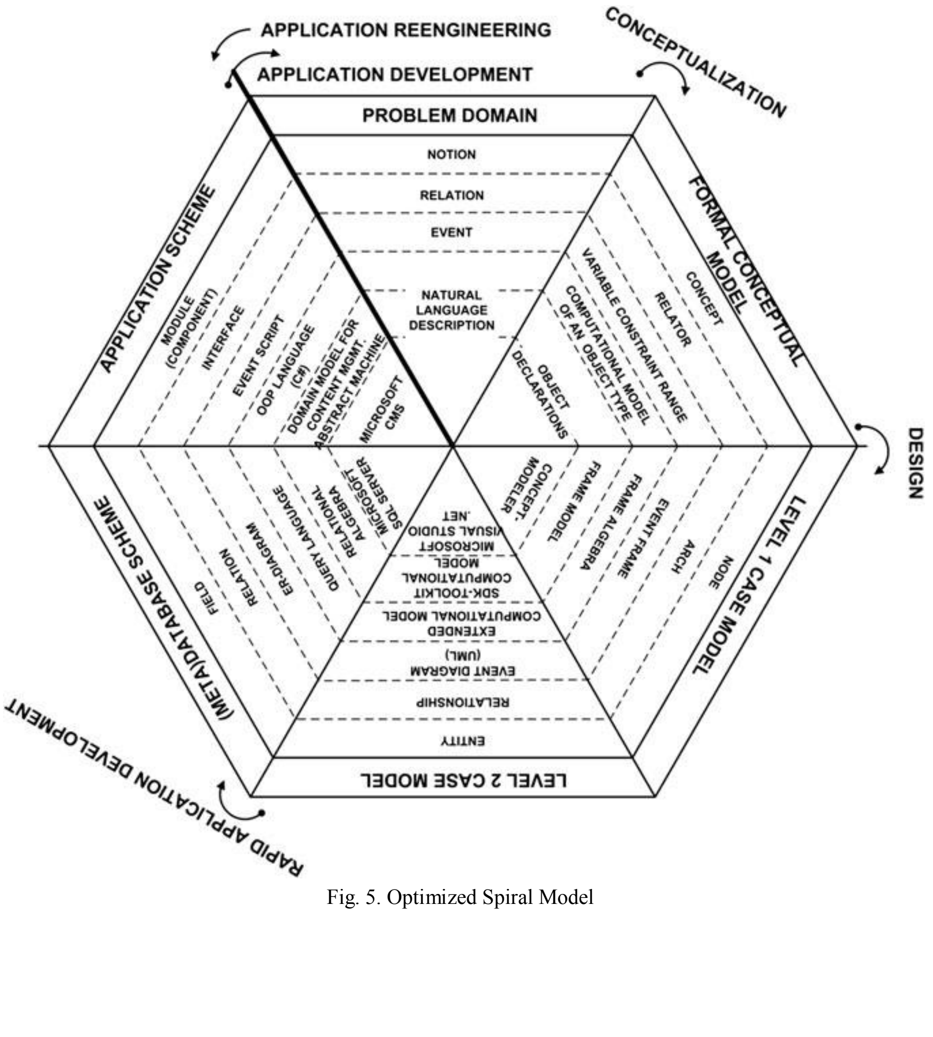
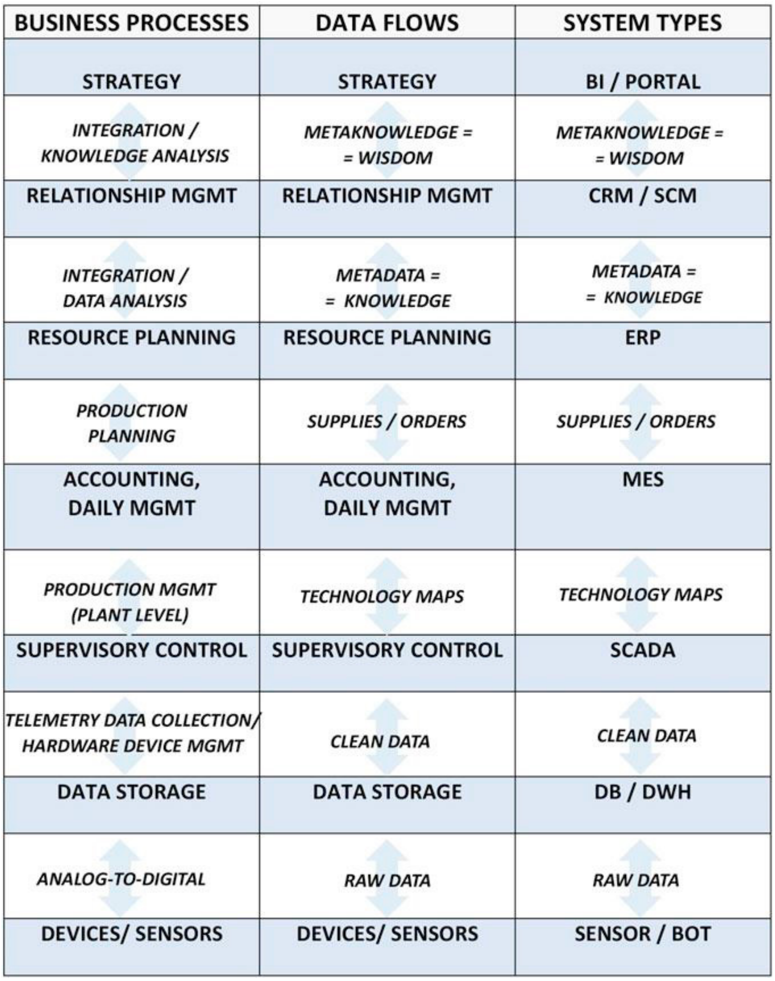


Fig. 5. Optimized Spiral Model

10. Agility Matrix

Agility is related to balancing business requirements and technology constraints. Many local crises result from misbalancing of these two aspects. Hence, agility is essential in other to achieve a well-balanced software solution. Agility is a remedy for crisis. When a crisis occurs, agility is vital for any kind of a business organisation, if it wants to overcome the crisis. Agility should be present in each and every stage of the enterprise system life cycle. In the case of an enterprise, its agility is a concept that incorporates the ideas of “flexibility, balance, adaptability, and coordination under one umbrella” [16]. For each of the perspectives, we identify a set of business management levels, such as strategic decision-making or everyday management. After we combine these perspectives and the business management levels, we get the enterprise agility matrix (Fig. 6); this matrix is a tool for monitoring, predicting and preventing the crises.



BUSINESS PROCESSES	DATA FLOWS	SYSTEM TYPES
STRATEGY	STRATEGY	BI / PORTAL
INTEGRATION / KNOWLEDGE ANALYSIS	*METAKNOWLEDGE = = WISDOM*	*METAKNOWLEDGE = = WISDOM*
RELATIONSHIP MGMT	RELATIONSHIP MGMT	CRM / SCM
INTEGRATION / DATA ANALYSIS	*METADATA = = KNOWLEDGE*	*METADATA = = KNOWLEDGE*
RESOURCE PLANNING	RESOURCE PLANNING	ERP
PRODUCTION PLANNING	*SUPPLIES / ORDERS*	*SUPPLIES / ORDERS*
ACCOUNTING, DAILY MGMT	ACCOUNTING, DAILY MGMT	MES
PRODUCTION MGMT (PLANT LEVEL)	*TECHNOLOGY MAPS*	*TECHNOLOGY MAPS*
SUPERVISORY CONTROL	SUPERVISORY CONTROL	SCADA
TELEMETRY DATA COLLECTION/ HARDWARE DEVICE MGMT	*CLEAN DATA*	*CLEAN DATA*
DATA STORAGE	DATA STORAGE	DB / DWH
ANALOG-TO-DIGITAL	*RAW DATA*	*RAW DATA*
DEVICES/ SENSORS	DEVICES/ SENSORS	SENSOR / BOT

Fig. 6. Agility Matrix

11. Conclusion

After 50 years, IT crisis, is still a critical issue. To address these crises, a specific discipline is required, and we suggest the IT Crisisology Framework (ITCF) to manage this issue. However, as material and IT product lifecycles are very different, to perform crisis management, there should be a combination of methodology-based optimization (e.g. Agile, SCRUM, XP, OpenUP, ICONIX) and communication. This results in the conquering of crisis as our ITCF approach is combined with software development discipline, knowing that factors determining efficiency and productivity in the short and long term are delivered and integrated according to a predictable model.

Our ITCF approach incorporates the three “pillars” of business, technology and human-related factors (T-, B- and H-factors); it also includes a set of lifecycle-based models. These models include data, process and system related aspects. One aspect of presenting and balancing the above three aspects in crisis is the Agility Matrix; to address data lifecycle specifically, we recommend using the Optimized Spiral Model. To manage technology (and knowledge) transfer, the authors suggest using enhanced Shannon information model and the “soft skills” tailoring based on the “seven principles”. Adizes framework and agile methods are also used to predict crises, and dynamically adjust the “iron” triangle parameters on order to optimize the lifecycle based on mission-critical trade-offs (ACDM/ATAM and SMART methods).

Systemic application of the above-mentioned models, methods, principles and practices under the umbrella of the ITCF methodology results in disciplined, predictable, manageable and adjustable software product development even in case of volatile and uncertain crisis conditions

References

[1] Colburn, A., Hsieh, J., Kehrt, M., & Kimball, A. (2008). There is no software engineering crisis. http://mercury.pr.erau.edu/~siewerts/cs332/documents/Papers/There-is-no-Software-Engineering-Crisis.pdf. Last accessed 23 April 2019.

[2] Buettner, M., Peng, D., Suporn, P., & Richa, P. (2006). Software Engineering is indeed in a crisis. https://www.researchgate.net/publication/265070905_Software_Engineering_is_indeed_in_a_crisis. Last accessed 23 April 2019.

[3] Adizes, I. Managing Corporate lifecycles – How Organizations Grow, Age and Die, Volume 1. Second edition. Santa Barbara, CA: *The Adizes Institute Publications*, in conjunction with *Embassy Book distributors*, 2012.

[4] Adizes, I. Managing Corporate lifecycles – Analyzing Organizational Behavior and Raising Healthy organizations, Volume 2. Second Edition. Santa Barbara, CA: *The Adizes Institute Publications*, 2015.

[5] Zykov, S. V. (2018). Managing Software Crisis: A Smart Way to Enterprise Agility. Issue 92: Springer Series in Smart Innovation, Systems and Technologies. Switzerland: *Springer International Publishing*, 2018, 153pp.

[6] Myers Briggs. Type Indicator. https://www.researchgate.net/publication/232553798_The_Myers-Briggs_Type_Indicator_and_leadership. Last accessed 23 April 2019.

[7] NATO Science Committee (1968). Software Engineering. http://homepages.cs.ncl.ac.uk/brian.randell/NATO/index.html. Last accessed 25 January 2019.

[8] Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., et al. (2001). Manifesto for Agile Software Development. http://agilemanifesto.org. Last accessed 25 January 2019.

[9] Smart Sheet (2019). Agile Methodology. https://www.smartsheet.com/agile-vs-scrum-vs-waterfall-vs-kanban. Last accessed 27 January 2019.

[10] Maximini, D. (2015). The Scrum Culture. https://www.springer.com/ us/book/9783319365053#otherversion=9783319118260. Last accessed 27 January 2019.

[11] Tutorials Point (2019). Extreme Programming. https://www.tutorials point.com/extreme_programming/index.htm. Last accessed 27 January 2019.

[12] Tutorials Point (2019). Agile Methodology. https://www.tutorialspoint. com/agile/index.htm. Last accessed 27 January 2019.

[13] Tutorials Point (2019). Adaptive Software Development. https://www.tutorialspoint.com/adaptive_software_development. Last accessed 30 January 2019.

[14] Rehkopf, M. (2019). Kanban. https://www.atlassian.com/agile/kanban/ boards. Last accessed 30 January 2019.

[15] LeanKit (2019). Principles of Lean Development. https://leankit.com/ learn/lean/principles-of-lean-development. Last accessed 30 January 2019.

[16] Extreme Programming (2013). Extreme Programming Project. http://www.extremeprogramming.org/map/project.html. Last accessed 30 January 2019.

[17] Mrsic, M. (2017). Crystal Method. https://activecollab.com/blog/project-management/crystal-methods. Last accessed 30 January 2019.

[18] Kuchins, A. C., Beavin, A., & Bryndza, A. (2008). Russia’s 2020 strategic economic goals and the role of international integration. Washington, D.C.: *Center for Strategic and International Studies*.

[19] Zykov, S. V. (2016). Crisis Management for Software Development and Knowledge Transfer. Issue 61: Springer Series in Smart Innovation, Systems and Technologies. Switzerland: *Springer International Publishing*, 2016, 133pp.

[20] Victor H. Mair (2009) http://www.pinyin.info/chinese/crisis.html. Last accessed 24 April 2019.

[21] Steve Nguyen (2014) https://workplacepsychology.net/2014/08/10/in-chinese-crisis-does-not-mean-danger-and-opportunity. Last accessed 24 April 2019.