

Customer Centric Project Management

Engaging stakeholders is not just to manage changes to requirements

Charles Villanyi Bokor

The CERP Group, Ottawa, Canada

ABSTRACT

Customer Centric Project Management (CCPM) is defined as the continuous re-examination, evolution and integration of organizational strategy, desired outcomes, stakeholders' vital needs and expectations, business processes, and project development methodology, into the enabling technology project's mandate and deliverables. It is to produce deliverables/outputs that are fit for the purpose and can be leveraged to achieving the intended outcome.

CCPM is based on empirical observations and current literature on project management and requirements change management, as well as on limited trials and test. When approved by executives, CCPM engages stakeholders to define the outcome and continuously during the project development lifecycle, enhance the vital requirements that define the output to be produced. It is often viewed by IT as involving '*them*' in what '*we*' do. CCPM focuses less on *how* we meet estimated schedules or costs and more on *what* are the vital outputs so corporations can create business value. It is not new, but this return to the obvious requires a cultural shift, and holds the involved senior executives, system owners, business analysts and the project manager, accountable.

Key Words: project development, customer centric, business requirements, requirements definition process, project management, requirements change management, project development team, outcome, methodology, value, accountability, project failure.

THE PROBLEM

In a recent article on the "10 common causes of IT project failure" [Carlson, 2013] one of the causes was: "Letting users delay projects by constantly requesting tweaks". According to this cited cause of project failure and common belief, allowing business requirements to be changed post system design (that is based on the requirements) and during project development, i.e. allowing tweaks, may force the designed and partially developed project to be reworked, delaying the project and or adding costs that were not budgeted. A 1996 article [McConnell, 1996] noted that: "Studies have found that reworking defective requirements, design, and code typically consumes 40 to 50 percent of the total cost of software development (Jones 1986)". Such significant variance between the pre-design estimated and the eventual cost of delivery or

time to deliver, can in turn fail and have failed IT projects. Thus according to this school of thought changes to requirements should be limited to minimize changes to schedules and costs.

Limiting customer requested ‘tweaks’ or changes to the requirements is considering the changes to be less important than the originally stated requirements. It implies that the customer was not aware that the requested changes can be postponed. Not easily allowing changes to requirements implies that the customers who presented the original requirements and the Business Analysts (BA) who elaborated them and consequently defined the project’s output, knew the right or best business solution with which to create the expected business value or outcome, which is the original (not new) and the future definition of project success. It implies that they have expressed *all* vital customer business needs and expectations at the time the requirements were gathered and defined. It further implies that since the time the requirements were defined no one has learned or identified a better way to solve the business problem. In other words, due to an organizational flaw that prevents the organization from innovating or naivety there has not been any learning during the development of the project. By keeping the list of requirements static, the organization implicitly declares, that those who can prevent the additional effort to develop the changes or tweaks or act on opportunities to improve needed capability to be made, can decide on the eventual business results. By not making changes the mandated functionality is delivered over the significantly higher cost of an enhanced system indicating that being on-time, on-schedule and / or within budget is the organization’s priority.

People still exist who believe that a project should and could get all user requirements documented [waterfall concept] before project design and development. Some people still believe that people, organizations or project teams can if they try, involve the client/customer business experts and produce the list of requirements that does not need any further changes.

IT spending in Canada is in the neighborhood of \$270 billion per year based on it being \$2.68 trillion in the US [NY Times, 2013]. The annual cost of IT project failures is \$1.2 trillion in the US and \$6.2 globally [Sessions, 2009]. As a generalization, the lifetime cost of an application is about 6.7 times [Outsystems] its initial development cost and some or most of this cost is to fund what many in the profession term “PHASE II”, which is the ensuing phase to develop the needed requirements that were not delivered in Phase I. We know and now must practice how to develop projects that deliver the right functionality for a fraction of the cost we spend on them today.

CUSTOMER CENTRIC PROJECT MANAGEMENT

Customer Centric Project Management (CCPM) is continuous aligning the skilled re-examination and readjustment of the technology project’s mandated business based deliverables, customers’ expectations and capabilities, the project’s estimated budget, schedule, and the system’s functionality as well as the cost of not making or postponing the identified change. It is ensuring that stakeholder needs, and options are evaluated to determine balanced, agreed-on enterprise objectives to be achieved. It is setting direction through prioritization and decision making and monitoring performance to satisfy customers more than to better manage the project.

It is the critical activity of engaging customers, for whom the project is being developed, as business requirements subject matter experts (SME), members of the project development team and input authority in the project's design.

According to a World Quality Report [WQR] commissioned by HPE, Capgemini and Sogeti, quality assurance (QA) is more important than ever to 1,560 executives surveyed. Top priority of these respondents was focusing on what is of value to customers. As such CCPM aligns with International Standards (9001: 2015), Customer Focus (5.1.2.) which is the top most item in 7 Quality Management Principles (QMP). It is an enhancement of the defective and out of date requirements change definition process commonly used today. Its aim is to define the vital deliverables (functionality) or the development project's necessary output that customers want, expect and will use, based on the latest understanding of the organizational strategy, business processes, capacity and capability. CCPM is about prioritizing customer satisfaction due to delivered system functionality that facilitate achieving organizational goals. ISO 9001.2015 requirement is to: "...*determine and select opportunities for improvement and implement necessary actions to meet customer requirements and enhance customer satisfaction...*".

CCPM is not a Holy Grail that will fix all the many problems that fail projects, but it is a solution to a specific and significant project development problem that occurs too often and has too big an impact on organizations.

STAKEHOLDERS

CCPM project management methodology involves a number of stakeholders including the organization's executives, the Project Manager (PM), the customer, program or system owner who funds and is to exploit the system, and the Business Analyst (BA) who is to facilitate customers to define their vital requirements. Each has a specific role and accountability.

In top performing organizations [HBR, 2002] senior executives are responsible for reengineering processes that use IT systems and for generating business value. Executives are implicitly accountable for project success, and must authorize the use of CCPM as it changes how changes to requirements are managed. If executives do not 'buy-in' to this methodology, it will not be possible to use it. However, choosing to not allow CCPM has to make them accountable for the missed opportunity cost of not having or postponing the deployment of newly discovered vital requirement(s) that would have been deployed using CCPM.

If executives do 'buy-in', they have to elaborate and communicate their support to the customer, the BAs and PM involved, and describe how the impact of continuous changes to requirements and consequently the budget and the schedule will be handled (governance).

The customer(s), sometimes referred to as business partner, must be and is held accountable for specifying the right thing that needs to be done and funds the project as well as any costs resulting from the requested changes to the requirements. This includes the return on the

investment spent on making the changes to the requirements, specifying the outputs the project needs to deliver and attaining the desired outcome with or without the requested changes that is determined at the post mortem analysis of accomplishments.

The PM is the third stakeholder in CCPM. PMs generally are trained and skilled in doing things right, i.e. planning and controlling system development to deliver projects efficiently and in spite of the many organizational constraints in their way. It is why they have been chosen to lead and manage the development and system implementation (not deployment) processes and entrusted with corporate resources.

In most organizations PMs are ill-equipped to balance customers' wishes for changes to requirements and currently used on-time and on-schedule development project success criteria that is focused on efficiency. However, CCPM makes it easy and convenient for customers to ask for tweaks, for PMs to listen (hear) to customers and follow up and acknowledge their input and requests. CCPM allows the PM to manage the development process as effectively (not only efficiently) as the organizational capacity, capability, executive management and their own expertise allows.

Peter Drucker is quoted to have said: "There is nothing so useless as doing [developing] efficiently that which should not be done at all.", which may be one of the reasons for the CHAOS Manifesto [CHAOS Manifesto 2013] mentioned later on. These are some of the reasons why the BA is the fourth and final stakeholder. BAs are the SMEs who must ascertain and then define and translate the customers' business and vital business needs the customers' expectations, and the mandatory business and regulatory requirements, as well as all considered alternatives. They in addition must also underline and incorporate into the business solution, any existing opportunities that align to the organization's strategy, but within the organization's capacity and capability. Then they have to facilitate changes to system requirements to be made as requested. This not only elevates the need for qualified BAs but also the role that the BAs can and needs to play. On the other hand, BAs must assist the CIO by providing intelligence on the information, procedures, and business decision flows, and define and document proposed interfaces of manual to automated operations and establish acceptance test criteria.

To underline their importance, McConnell [McConnell. 1996] states that: "As a rule of thumb, every hour you spend on defect prevention [making changes to requirements] will reduce your repair time [by] three to ten hours. In the worst case, reworking a software requirements problem once the software is in operation typically costs 50 to 200 times what it would take to rework the problem in the requirements stage." This argues that BAs must spend more time to define requirements during design and 'tweaks' should be encouraged, allowed or done during project development, i.e. before not after implementation.

While today's change management, i.e. accepting or not accepting new requirements and changing the design of the system, is often about the cost of the changes, it often does not include the calculation of what the cost of not making the changes will be. Therefore, the BA

must be held accountable for the professional attempt to uncover how business is done in the organization and to identify needed work processes that do not strain the corporate culture.

THE PROPOSAL

CCPM is about continuously and more directly engaging stakeholders during the System Development Life Cycle (SDLC), in evolving the list of requirements which in turn define the final project deliverables. It is not constrained by the estimated time to develop, the established schedule or the estimated cost to deliver. It is often viewed by IT as involving ‘*them*’ in what ‘*we*’ do. CCPM is not a favored methodology of insecure PMs or risk averse organizational cultures, and is something that many who dislike accountability strive to avoid. Nor is it acceptable to executives, who prioritize delivering projects within the estimated budget, in an estimated amount of time, as per the planned schedule and with the original scope, to replace standard requirements change management processes with CCPM. For many, the customer-centric iterative requirements definition process is too great a shift of the culture. CCPM is for organizations where executives prioritize the value chain and defining ‘*what*’ value the system’s users will be able to create, not ‘*how*’ work is to be done.

CCPM is centered on the concept that projects are not developed to be on-time, on schedule or to deliver defined functionality. They are developed specifically for one of four reasons [Charles’ Law] and in general so that internal customers can more easily change business results (outcomes), the original, not new, re-discovered and the future definition of success. As such, customers must be more directly and continuously engaged and held accountable for specifying all vital requirements, for the cost of the requested requirements as well as for the impact the new system will have on the organization. Assuming that these requirements are written by BAs, implies that the BAs must be held accountable for elaborating and evolving business needs that can be embed into the socio-organizational environment as part of Organizational Change Management (OCM) needed for every large scale project deployment.

This approach conflicts with many executives’ views and project management practices (e.g. changing the cost of the project when a requirement is presented to be included). CCPM is not a different way to manage projects but an enhancement to how to change business requirements and how the development methodology integrates customers into the SDLC. As W. Edward Deming said: “It is not necessary to change. Survival is not mandatory.”

The concept of facilitating making changes to requirements during the project is not new, but a cultural shift. This shift requires the organization to prioritize *what* the project delivers, i.e. the output, and the eventual outcome it enables or facilitates over *how* close the final cost and schedule are to be to the estimated budget and wished for schedule. CCPM is not more efficient but a more effective Requirements Change Management methodology. Its focus is on the needs of the organization and the customers, but it is also inadvertently a better way to control total cost of application ownership.

As a general rule business requirements evolve after they are defined in the Requirements Definition Document. Much as a restaurant patron needs in real time to be able to order a second bottle of wine during the meal and change his/her mind about having or skipping desert, so too internal customers need to be able to make changes to requirements when they realize a previously unforeseen possibility. Hence customers need to be continuously engaged in the evolution of project requirement. And like the restaurant patron who has to pay for the additional bottle of wine, the customers' 'right to change' is tied to his/her accountability for the extra effort (corporate resources and capacity) expended to make the change(s).

A second and more controversial reason for CCPM comes from the CHAOS Manifesto [CHAOS Manifesto 2013]. It suggests that: "20% of features [of an application] are used often and 50% of features are hardly ever or never used. ...The task of requirements gathering, selecting, and implementing is the most difficult in developing custom applications. ... 20% of the features that give you 80% of the value ... Therefore, reducing scope and not doing 100% of the features and functions is not only a valid strategy, but a prudent one." It does not specify who should make the decision or how it is to be decided which part of the scope (features and functions) not to deliver. However, as the PM is focused on IT project delivery and IT has no domain knowledge (business expertise), the PM cannot make this currently practiced error and mistake. Arguably, only customers who know or should know how to create value for the organization, who are accountable for the costs incurred and who have proper incentives, can decide to pay for the benefits that requested changes to requirements promises to provide. Thus customers need to be engaged in the requirements definition, design and output definition process.

CCPM balances the imperatives of what the business needs according to established strategy, accountability for resources expended and what the technology and the project team can deliver. It integrates the goals of the corporation, the needs and expectations of the corporate customers with the capabilities of the project development team. At the same time CCPM allocates explicit accountability for project success to the four stakeholder groups. This idea of separating accountabilities is supported by Gartner [gartner.com] which claims that: "*Organizations that separate the 'doing the right thing' from 'doing things right' tend to be more successful in the long term.*"

Is CCPM important? Some indicators say it is. IT spending is in the neighborhood of \$270 billion / year in Canada based on \$2.68 trillion posted by NY Times [NY Times, 2013]. The annual cost of IT project failures is \$1.2 trillion in the US and \$6.2 globally [Sessions, 2009] and according to Gartner, 80-85% of IT spending is to keep the lights on, i.e. making changes after deployment. As a generalization, the lifetime cost of an application ends up being about 6.7 times [Outsystems] its initial cost of development and some or most of this cost is due to what many in the profession term "The PHASE II", meaning moving all the requirements that are needed but are not delivered to a subsequent phase. We need to learn how to develop systems that deliver the right outcome and hence, we need to learn how to make changes to requirements.

CONCLUSIONS

Organizations need to do a better job of evolving system requirements, as this weakness is costing corporations dearly and negatively impacts the relevance of IT. CCPM aims to make the customer as successful as possible, by engaging the customers who are accountable for the additional costs and delays or changes to the approved ROI caused by the changes to requirements. However, if CCPM is not to be a career changer (end-er) for the PM, then its use has to be approved by corporate executives prior to the start of the project. If executives have not approved it explicitly, then they must accept accountability for the opportunity lost.

CCPM separates the business expertise and those who must take accountability for business decisions from project management expertise and the skill and accountability to deliver to plan expectations. To satisfy and delight (from Six Sigma and Lean Manufacturing) the project customer, PMs must only be accountable for areas under their control and thus be able to change cost and elapsed time when new requirements are presented to be evaluated and or integrated into the project.

But CCPM is like an exercise bike in that if executives' effort is to explain why it does not work in their organization, in other words it is not used as the bicycle is not ridden, it will not improve anything.

REFERENCES

- [1] Caron Carlson, *10 common causes of IT project failure*, FierceCIO, April 3, 2013
- [2] Steve McConnell, CEO and Chief Software Engineer at Construx Software, 1996 article
- [3] Capers Jones, *Programming Productivity*, New York, McGraw-Hill, 1986
- M. Gibbs, NETWORKWORLD, May 9, 2013,
- [4] World Quality Report, Capgemini: Customer experience emerges as a top QA priority,
- [5] Harvard Business Review, *Six IT Decisions Your IT People Shouldn't Make*, Nov 2002,
- [6] CHAOS Manifesto 2013, *Think Big Act Small*, The Standish Group International
- [7] NY Times, *PMOs: One Size Does not Fit All*, 2013
- [8] Roger Sessions, CTO ObjectWatch, *The IT Complexity Crisis: Danger and Opportunity*, 2009 White Paper,
- [9] Outsystems, *Why IT struggles to innovate and how you can fix it*,

[Charles' Law] Software application projects are not designed, approved, developed and implemented to be on-time, on-budget and within scope. They are developed, as per Charles' Law, for one of only four reasons. These are:

1. To decrease operating costs - decrease or eliminate the existing expense or the rate of rise in costs, or alternatively decrease impediments that inhibit a more cost-effective way to operate (Note that cost can be replaced by resources, or inputs used to generate output. Hence the concept is valid for non-profit establishments);
2. To increase revenue – by increasing market share or gaining competitive advantage, or alternatively providing more customer focused service or better outcomes;
3. To keep someone out of jail - generally a person in the 'C' ('Chief' as in CEO, CFO, CIO) suite, by meeting legal and compliance requirements or applicable legislation; or
4. To boost a personal ego - as when a Project Champion uses his or her prestige or power to develop a project that may not be an organizational priority.

About the Author



Charles Villanyi Bokor

Ottawa, Canada



Charles Villanyi Bokor is a Strategic Management Consultant focused on Leading to Better Decisions. Principal activities include Business Transformation, Problem Project Recovery & Leadership, Strategic Planning. Charles works mostly in Ottawa but has successfully completed assignments in Florida, Wales, Malaysia, Sweden and Australia, and was key-note speaker in Johannesburg South Africa and Victoria BC. Formal education includes an Executive Development and Diploma in Management (McGill University), M.Sc. Mathematics (Université de Grenoble, and U. de Montréal) and B. Sc. Mathematics (Concordia University). He was: Program Director of the Corporate Performance Management Program, Spratt, Carleton; Director of IS/IM at Royal Trust; and at Northern Telecom; CMC; CMC Board Member; PMI-OVOC Board Member; Governor of ICCG; is ITIL Certified and a TBS Independent Project Reviewer.

Charles Villanyi Bokor can be contacted at villanyibokor@gmail.com