

Information Technology Project

A Challenge for Project Management

Tribhuvannath Mishra, PMP

tmishra@outlook.com

I. ABSTRACT:

With this paper we want to highlight life cycle of Information Technology (IT) project, Uniqueness of IT projects, success and failure of IT Projects. It lists and explains some of the most common reasons for IT project failure. It also helps in understanding some of the myths of failure of IT failures. It also provides some of the tips and tricks to avoid the common mistakes committed by project management.

II. INTRODUCTION:

After spending millions of dollars, the last thing you would like to hear from your top executive that your project is cancelled, halted or changed direction due to failure. You may start wondering what went wrong or what could have been done differently. It is very easy to blame others after a failure. A common practice is to award the culprit and punish the innocent. Based on who you ask, you may hear conflicting answer. Project manager may blame technical expertise of technical staff, lack of support from support services, delay from procurement, tight schedule, contingencies, HR department for not providing skilled resources. End users may blame requirement analyst, finance department, vendor selection process. Vendor may blame tight bureaucracy for delay and the list goes on and on.

When a project fails everybody from the team bears some sort of responsibility so finger pointing after project failure doesn't help. If you are facing IT project failure for the first time, just understand that you are not first one to be part of failed IT projects neither the last one. According to "Improving IT Project Outcomes by Systematically Managing and Hedging Risk," a 2009 IDC report [1] by Dana Wiklund and Joseph C. Pucciarelli, 25 percent of IT projects fail outright. Meanwhile, 20 to 25 percent don't provide ROI and up to 50 percent require material rework".

III. IT PROJECT LIFE CYCLE (PLC) OR SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)

SDLC consist of the various development stages or workflows of the software process. A software development life cycle consist of stages as below as per pap

- Requirements – defines needed information, function, behavior, performance and interfaces.

- Design – data structures, software architecture, interface representations, algorithmic details.
- Implementation – source code, database, user documentation, testing.
- Testing - Test against Requirement
- Deployment - Moving code to production
- Post Deployment Support - Resolve any issue after deployment.

There are various frameworks for SDLC. Some of them are listed below

- Waterfall SDLC Model – Distinct stages as above planned in sequence

Advantages: Easy to understand, requirements stability and good for management control for planning, staffing and tracking purpose.

Deficiency: Not flexible for change, Customer review at the end put project at risk if requirement is not properly understood and signed off.

Hard to implement if requirement is not stable or demand is continuously changing as in many IT projects.

- Spiral SDLC Model–Rapid prototyping model plus risk analysis preceding each phase

Advantages: Early and frequent feedback from users, Cumulative costs assessed frequently, Provides early indication of insurmountable risks, without much cost

Deficiency: Developers must be reassigned during non-development, May be hard to define objective, verifiable milestones that indicate readiness to proceed through the next iteration phase activities, • Time spent planning, resetting objectives, doing risk analysis and prototyping may be excessive

- Incremental SDLC Model – Construct a partial implementation of a total system. Later add increased functionality

Advantages: Customers get important functionality early. Each release delivers an operational product with added functionality. Customer can respond to each build result in lower risk of requirement change.

Deficiency: Requires early definition of a complete and fully functional system and definition of

increments. User and developer should understand the functionality of each modular release to avoid mismatch of requirement and delivery.

- Agile SDLC Models [6]- Agile development offers a lightweight framework for helping teams to deliver rapid solution with focused scope in constantly evolving functional and technical landscape. Dynamic Systems Development Method (DSDM), Scrum , Extreme Programming (XP), Rational Unify Process (RUP) are some of Agile SDLC Model. Each one of these bypass one or more standard life cycle process. This is used in disciplined environment with less formal setup. Some of them are as below
 - Dynamic Systems Development Method (DSDM): Created with goal of devising and promoting a common industry framework for rapid software delivery. Requirements are baseline at a high level early in the project. Rework is built into the process, and all development changes must be reversible. Requirements are planned and delivered in short, fixed-length time-boxes, also referred to as iterations, and requirements for DSDM projects are prioritized using MoSCoW Rules: M – Must have requirements, S – Should have if at all possible, C – Could have but not critical, W - Won't have this time, but potentially later
 - Scrum: The "Product Owner" works closely with the team to identify and prioritize system functionality in form of a "Product Backlog". The Product Backlog consists of features, bug fixes, non-functional requirements, etc. - whatever needs to be done in order to successfully deliver a working software system. With priorities driven by the Product Owner, cross-functional teams estimate and sign-up to deliver "potentially shippable increments" of software during successive sprints, typically lasting 30 days.
 - XP (Extreme Programming) features Setting up of business priorities, small releases, simple design , continuous unit testing by programmer, continuous user acceptance testing(UAT) by Users, Continuous Refactoring, Collective ownership of code, Continuous integration, On-site customer and high coding standards

Advantages: Agile development accelerates the delivery of initial business value, and through a process of continuous planning and feedback, is able to ensure that value is continuing to be maximized throughout the development process. As a result of this iterative planning and feedback loop, teams are able to continuously align the delivered software with desired business needs, easily adapting to changing requirements throughout the process.

Deficiency: Agile methodology may not suit for legacy system, Hard to plan workforce due to iterative approach. Budget allocation may be very difficult. Well-disciplined environment require to achieve goals in short terms.

IV. DEFINITION OF PROJECT FAILURE

A project is considered a failure when it has not delivered what was required, in line with expectations. There are various definition of project failure and success. Simplest one uses project management triangle i.e. a project must deliver to the scope, to cost, to quality, and on time; else project considered a failure.

As per PMBOK 4.0 guide from PMI [4] scope, schedule, budget, risk, resources and quality needs to be monitored and controlled to avoid project failure. Effective communication is another area which need to managed properly throughout project for successful project . As per Robert Diana in his paper Defining Project Failure [3] If key stakeholders agreed that a project had to exceed its initial budget, the project may still be considered a success. Likewise, if a project delivered everything that was in the detailed project designs, it may still be considered a failure if it didn't include vital elements that the key stakeholders needed. This doesn't seem fair, but project success and failure isn't just about the facts, nor is it simply about what was delivered. It's also, crucially, about how the project is perceived. Reason communication particularly to stakeholders is big part in project success or failure.

Paper named Five Areas Influencing Project Success or Failure published in IBM System Magazine [2] list documented causes of IT project failure, a majority—54 percent—are attributed to project management. Surprisingly to some, technical challenges are the least-cited factor at 3 percent while business, people and method related failures attribute to rest.

V. UNIQUENESS OF IT PROJECTS

IT projects are unique compared to other projects such as building construction, retails, healthcare, education, roadways, railroad, shipbuilding, aircraft building, Chemical engineering, petroleum pipelines, robotics, automobile and many more. Some of the basic characteristics of IT projects are listed below.

Changing technology: IT projects are always at risk due to fast changing technology. New software, hardware and methodologies bring new challenges every day. Most of the projects which tries to be ahead of curve in this changing technology has risk of facing lot of unknown problems while others who are too late to try a technology may face obsolete technology before even completion of project. a company dealing with student enrollment tried to upgrade their system at the end of support before even

project was in planning face new version of software already became a norm in the industry. Project was halted halfway after spending couple of million dollars on obsolete technology and new project scope was created for next version of product.

Short Life Cycle: Most of software projects are short life cycle due to fast changing technology and changing demand. Even Large projects are broken down into smaller projects to mitigate risk. Typical IT projects are from 3 months to 12 months in durations. Integration of various projects plays a major role in project. Most of the short projects has no or very little wiggling room for schedule.

Overlapping of Projects: Any IT projects require human resource of varied skill and knowledge. It's hard to have dedicated personnel for each project. Lot of IT Projects run in parallel with conflicting schedule. Resources with different skill set are shared in multiple projects. Sometime a project may take 2 weeks to finish but it has to wait for another project team to finish its part so that changes can be integrated by a third project team member.

People Dependent: Each IT project requires different skill sets and knowledge for team members. Due to short duration of projects, it's very difficult to hire new people with new skill. There are always a balance approach need to be taken for what existing personnel can do and what can be subcontracted. Availability of short term and long term budget also plays role in hiring decisions. A chunk of total project cost is attributed to people. If resources are not managed properly project budget can easily overrun in very short period of time.

Project Management: Managing IT projects is an art in itself as projects manager has to keep up himself in terms of changing technology but also has to manage regular project management soft skills. A cunning team member can misdirect a project manager in wrong direction resulting in project failure. A technical stakeholder can cause scope creep if project manager fail to understand technology properly. Lot of IT project fails at planning stage itself because project manager focus too much on visible risk, schedule centric to delivery date and headcounts while failing to understand complexity of technology. Without technology knowledge project manager may fail to put people with right skill set and right contingency plan in right place. One of the ERP project run green till almost near finish line and then realized that some of the basic functionality couldn't be fulfilled. The project manager kept ignoring the initial warnings by some of the more experienced team member in favor of vendors promise to deliver. Millions of dollar and human efforts could have been easily saved with little more technical detail exploration at the beginning of project. In one of the health care project, the project manager failed to understand technical expertise provided by one of team member and let him go in favor of person with better communication skills resulting in failing of 2nd phase of the project due lack of required technical skill.

VI. COMMON MYTHS ABOUT IT PROJECT

Even though most of IT projects have some of the common characteristics list earlier but failure of IT project mayn't be pin pointed to some of the myth developed over the years. Some of myths listed below with some tips to avoid failure

- Lack of Technical competencies of team is main cause of IT Project failure

Most of IT project fails because of bad planning, lack of coordination, lack of communication than lack of technical competencies of team. There is and there will be argument on both side of aisle, if project manager requires domain knowledge or not. As per article in computer weekly [16] 75 % project fails because of bad planning while 69 % of project which failed had poor communication. In IT project new technology always presents new challenges but it can be accessed with smart project manager, good team and address effectively at planning stage. Some of the technical deficiency which get revealed during last stage of project are mostly uncovered as risk by team members initially but generally those sign ignored by aggressive project manager resulting in show stopper at later stage. Most of these problems can be addressed smartly at initial stage of project and some workaround can be developed, personnel can be trained or scope can be adjusted to prevent outright failure at later stage.

- Project Manager without Domain Knowledge can manage the IT project

There is and there will be arguments on both side of aisle, if project manager requires domain knowledge or not. Even though a good project manager who knows art of project management well and can learn the change quicker, domain knowledge at initial phase of project is not primary requirement but a good project managers with some domain knowledge has always better success rate over the project manager totally new to the domain. Project Manager with domain knowledge can have better understanding of what works and what doesn't at initial stage of project. Proper contingencies and resources can be planned well ahead at planning stage itself. A project manager with medical domain knowledge will know at very beginning to put best resources at claim handling. She/he will also know importance of proper handling of Personal Identifiable Information(PII) and medical records and time required to mask, scramble this data, formalities needs to be done before handing over data in lower environment. A project manager with domain knowledge of mortgage industry will plan for seasonal cycles and Sarbanes Oxley framework requirements. Project manager planned testing during winter which missed important spring seasonal surge of data resulting in performance issue when system was

delivered to user. One of Project manager missed key performance issue with budgeting module without having the prior domain knowledge and project was abandoned as issue surfaced at last stage of project.

- Working overtime can make up for initial Planning deficiencies[15]

Overtime can't be built into planned resource and schedule. Overtime can be used some time to meet the contingencies in the project but not on regular practice and should not be planned to meet the goal. In case resources are restricted scope should be managed well to prevent future failures. When people put extra hours continuously, it may eventually start affecting the quality of output and increase rework. Programming required lot of intellectual work and unrested brain may not able to create quality work. For short term if person is motivated enough can work extra to meet the demand but ultimately It will result in lower quality and productivity of individuals.

- Micro Managing Project can result in better efficiency[14]

Micromanagement start most often from the manager who loses focus on broad outline of project and try to direct self-frustration on employee who is not responsible for mess created by unrealistic performance goal, own insecurity, or as a tool to terminate an employee. Most of the time it doesn't create any positive result but employees start shutting down, stop giving positive feedback and sometime leave the stressful workplace. As long as employee is delivering what is expected under reasonable working conditions, micro management is unnecessary and should be avoided. Most of the micromanaged project leads to failure as communication become one way and creative ideas can't be expected from micro managed employees while micro managing manager loses focus on bigger picture of project.

- Project team learn the lessons from past implementation failures:[13]

Due to resistance to change, organizational culture, people fail to accept failure in past and organizations keep doing the same thing but expecting different results. A project team need to always conduct an independent implementation history assessment for similar project in the past so that they can learn from the past and identify what the organization did correct in past and need to continue or what needs to be done differently to avoid failure. Some team members and managers always end up doing the same mistakes every time they are involved in a project. These people needs to be trained properly before putting on any new project or there should be proper change agents

assigned to manage new changes to avoid repeat the failure.

- Outsourcing the work can resolved internal problems:

One company failed twice implementing ERP installation hired a 3rd party vendor to complete install for 3rd time. Project failed 3rd time as vendor selection process has some drawback which resulted in selecting a incompetent vendor who didn't had required expertise to implement the change. Project was managed by incompetent people internally who end up doing similar mistakes committed during first two implementation. There were other internal factors which caused failure of project in the past such as resistance to change, individualistic team members, one way communication with no feedback mechanism, improper planning for contingencies, unnecessary micromanagement and unmotivated staff resulted in to failure of project again even if some of technical problems were tried to be outsourced through external vendors. While selecting outsourcing partner if proper communication channels, checks and monitoring is not properly established , shifting of responsibilities, integration problems, insufficient follow up or tracking to allow issues to be surfaced and addressed early resulting in issue end up on critical path later. An external thorough analysis of past projects, mistakes done in the past and lesson learned can help to avoid these costly mistakes.

VII. CONCLUSION:

IT projects are unique. Most of these project are technology driven and can easily end up in failure if not managed properly. It project can be called a failure not only when expected results are not delivered within expected time, money and of proper value but also when its values are not perceived properly by stakeholders. .With shorter life cycle IT project presents challenges for project planning, staffing and tracking. New methodology such as Agile, XP, RAD are providing new tools for managing IT projects. A smart project manager with good domain knowledge, motivated competent team members, good communication methods and proper project control mechanism can reduce the chances of failure for an IT project. A prudent analysis of lesson learned from past failure of similar projects can also help to avoid the similar mistakes.

VIII. REFERENCES

- [1] Improving IT Project Outcomes by Systematically Managing and Hedging Risk," a 2009 IDC report [1] by Dana Wiklund and Joseph C. Pucciarelli
- [2] http://www.ibmssystemsmag.com/power/Systems-Management/Workload-Management/project_pitfalls/project_success_or_failure/
- [3] <http://agile.dzone.com/news/defining-project-failure>

- [4] PMBOK IV guide from PMI
- [5] http://www.ibmssystemsmag.com/power/Systems-Management/Workload-Management/project_pitfalls/
- [6] <http://www.versionone.com/Agile101/Agile-Software-Development-Benefits/>
- [7] http://www.mindtools.com/pages/article/newPPM_58.htm
- [8] <http://agile.csc.ncsu.edu/SEMATERIALS/AgileMethods.pdf>
- [9] <http://www.informationweek.com/global-cio/interviews/why-tech-projects-fail-5-unspeakable-reason/240152282>
- [10] <http://www.certmag.com/read.php?in=1394>
- [11] http://www.computerworld.com/s/article/9116470/IT_s_biggest_project_failures_and_what_we_can_learn_from_the_m?taxonomyId=73&pageNumber=4
- [12] http://www.umsl.edu/~sauterv/analysis/6840_f03_papers/frese/
- [13] http://www.imakenews.com/imaworldwide/e_article001680694.cfm?x=bgW0R13,bj7NbMRQ,w
- [14] http://www.ncmahq.org/files/articles/ecb0a_cm0707_c01.pdf
- [15] <http://www.facstaff.bucknell.edu/ttoole/Toole%20PM%20Causal%20Loop%20Diagram.pdf>
- [16] <http://www.computerweekly.com/feature/Why-projects-fail>

IX. ABOUT THE AUTHORS

Tribhuvannath Mishra, PMP, is a successful project management professional with Engineering degree. He has decades of experience in evaluating, designing, development, and deployment customize Data warehousing, Reporting, Archiving and ETL solutions for Government and Corporate clients in Retails, Manufacturing, Telecommunication, Healthcare, Insurance, Finance, and Education sector based on their specific needs. When he finds spare time, he loves to write poetry and book.