The failures of IT projects. Factors and early prediction.
Abstract

A number of researches and case studies has been collected and analysed with the mission to investigate the current situation where for the most part the IT project was considered failed or challenged. In the last paragraphs we elaborated on some ideas as to how to detect the failures in the early stages of a project.
1. Introduction

It is a well-known fact that many Information System projects have failed. (Ibrahim, Ayazi, Nasrmalek, & Shamin, May - June, 2013). Moreover, based on the number of researches, we should state that IS projects are still more likely to fail than not and this also includes large and national (government) projects.

In this paper, we collected a number of latest researches, which clearly indicate that the situation in IS projects as well as in IS project management is unlikely to improve. Furthermore there is a tendency to conceal some problematic issues and to not share the information regarding issues outside the team and other people involved in the project.

In 2005 Dr. Paul Dorsey in his report “Top 10 Reasons Why Systems Projects Fail” has named the current situation when 50-80% of large Information system projects frequently fail as a catastrophe. (Dorsey, 2005). Unfortunately nine year later we still cannot see significant improvement but we have more information gathered over a longer period of time to find out how to predict failures.

However, this report collects not just the disastrous statistics. Further investigation has allowed us to reveal a number of factors responsible for possible failures of IS projects. Those factors were considered in relation to the project environment, nature, ecosystem size and other parameters.

The awareness for those factors is the cornerstone of methodologies permitting predictions of theoretical probability of failure during the early stages of a project. Early prediction is the only way allowing to resolve the issues or to abandon the project before it becomes too late, before the investment will be spend on potential failure which is possible to predict.

A set of recommendations would be cited in the conclusion to assist project managers in identifying projects that are likely to fail. Some recommendations are obvious but there are also some seemingly unpredictable ideas for project managers.

* * *

The concept of Information System used in the report “originally includes the combination of computing hardware, communication technology and software” and “designed to manipulate information related to certain business processes”. (Ibrahim, Ayazi, Nasrmalek, & Shamin, May - June, 2013)
2. Background

The research of Standish Group analysing over 9000 IS projects completed in 2006 indicates that only 35% of the projects were successful, 19% were cancelled before completion. 46% of completed projects were completed over budget, late or with poorer functionality than initially declared. (Schach, 2011). In 2008 the percentage was not changed significantly. The “Chaos Report by Standish Group” is pictured the IS projects performance over a decade and the main outcomes of report was aggregated into the table below:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Succeeded (%)</td>
<td>16</td>
<td>27</td>
<td>26</td>
<td>28</td>
<td>29</td>
<td>35</td>
<td>32</td>
</tr>
<tr>
<td>Challenged (%)</td>
<td>53</td>
<td>33</td>
<td>46</td>
<td>49</td>
<td>53</td>
<td>46</td>
<td>44</td>
</tr>
<tr>
<td>Failed (%)</td>
<td>31</td>
<td>40</td>
<td>28</td>
<td>23</td>
<td>18</td>
<td>19</td>
<td>24</td>
</tr>
</tbody>
</table>

(Ibrahim, Ayazi, Nasrmalek, & Shamin, May - June, 2013)

Another investigation has been made for 214 IS projects which took place across EU between 1998 and 2005. Interviews were conducted with the project managers to collect the information on issues in those projects. To reach a clear result this investigation was made for selected IS projects from different industry areas.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Sector</th>
<th>No. of projects examined</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Manufacturing</td>
<td>43</td>
</tr>
<tr>
<td>2</td>
<td>Retail</td>
<td>36</td>
</tr>
<tr>
<td>3</td>
<td>Financial services</td>
<td>33</td>
</tr>
<tr>
<td>4</td>
<td>Transport</td>
<td>27</td>
</tr>
<tr>
<td>5</td>
<td>Health</td>
<td>18</td>
</tr>
<tr>
<td>6</td>
<td>Education</td>
<td>17</td>
</tr>
<tr>
<td>7</td>
<td>Defence</td>
<td>13</td>
</tr>
<tr>
<td>8</td>
<td>Construction</td>
<td>12</td>
</tr>
<tr>
<td>9</td>
<td>Logistics</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>Agriculture</td>
<td>6</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>214</td>
</tr>
</tbody>
</table>

(McManus & Wood-Harper, 2008)

The previous research of the same authors indicated that 7 of 10 projects in 2002 in UK adopted waterfall method. (McManus & Wood-Harper, 2008) The table below has been designed using the knowledge of waterfall SDLC (and its clones) stages and shows the result of the analysis in which less than 24% of projects were cancelled on the different stages of SDLC and almost “one in three were schedule and budget overruns “. (McManus & Wood-Harper, 2008)
The failures of IT projects. Factors and early prediction

<table>
<thead>
<tr>
<th>Lifecycle stage</th>
<th>Number of projects cancelled</th>
<th>Number of projects completed</th>
<th>Number of projects overrun (schedule and/or cost)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feasibility</td>
<td>None</td>
<td>214</td>
<td>None</td>
</tr>
<tr>
<td>Requirements analysis</td>
<td>3</td>
<td>211</td>
<td>None</td>
</tr>
<tr>
<td>Design</td>
<td>28</td>
<td>183</td>
<td>32</td>
</tr>
<tr>
<td>Code</td>
<td>15</td>
<td>168</td>
<td>57</td>
</tr>
<tr>
<td>Testing</td>
<td>4</td>
<td>164</td>
<td>57</td>
</tr>
<tr>
<td>Implementation</td>
<td>1</td>
<td>163</td>
<td>69</td>
</tr>
<tr>
<td>Handover</td>
<td>None</td>
<td>163</td>
<td>69</td>
</tr>
<tr>
<td>Percentages</td>
<td>23.8%</td>
<td>76.2%</td>
<td></td>
</tr>
</tbody>
</table>

(McManus & Wood-Harper, 2008)

Those tables based on research in EU have confirmed one more time the fact that “only one in eight information technology projects can be considered truly successful”. (McManus & Wood-Harper, 2008).

Obviously the successful project meets: “its budget, delivery and business objectives”. Moreover a project is concluded as failed if it either has “been cancelled or does not meet its objectives”. (Procaccino, Verner, & Overmyer, 2000) So in the table the 23.8% of cancelled projects is also the point of analysis based on the fact that cancellation in early stages (before the implementation phase) could be considered a fault but possibly still protect the main part of the investment.

For the large and national projects this percentage of success is even less optimistic. In the first decade of the 21st century about $3 trillion was used for e-government IS development. However, about 60-80% of those projects failed. (Stanforth, 2010)

ERP systems implementation is the abundant type of IT project since the end of the nineties when the companies began to prepare their systems for the problem of year 2000. System integrator, Panorama Consulting Solutions, has made the ERP research survey for the period between September, 2012 and January, 2013. This research was made mostly for companies with revenues of less than $300 million (71% of respondents) which means “this is primarily a small business survey” (Krigsman, 2013). So the result of this research is actually more interesting because it was designed for another part of market.

Summarized results showed that:

- over 50% of projects experienced cost overruns
- over 60% of projects experienced schedule overruns
- 60% of respondents “received under half of the expected benefit from their ERP implementation” (Krigsman, 2013)

In comparison with the previous researches below we could see that the picture hasn’t significantly changed.
Moreover those results are quite consistent with the other researches for significant and national (government) IT projects.

Again, there are three factors of failure: over budgeting, over estimation and low user satisfaction (quality is lower than expected).

Based on the Panorama Consulting Solutions report, most of ERP implementations “run late, over-budget, and do not deliver planned results” (Krigsman, 2013). However, only 10% of respondents actually named this result as a failure. That indicates only that users’ expectations are too low in this sector of the economy and the real situation must definitely be even worse.

In 2014 Panorama Consulting Group continue its investigation and made a new ERP report regarding the situation in ERP implementation market for the period from January 2013 to February 2014.

Table below indicates data summary from latest 4 reports where we can see an unfavourable tendency.

<table>
<thead>
<tr>
<th>YEAR</th>
<th>COST</th>
<th>% OF COST OVERRUNS</th>
<th>DURATION</th>
<th>% OF DURATION OVERRUNS</th>
<th>% RECEIVING 50% OR LESS BENEFITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>2012</td>
<td>$7.1MM</td>
<td>53%</td>
<td>17.8 months</td>
<td>51%</td>
<td>60%</td>
</tr>
<tr>
<td>2011</td>
<td>$10.5MM</td>
<td>56%</td>
<td>16 months</td>
<td>54%</td>
<td>48%</td>
</tr>
<tr>
<td>2010</td>
<td>$5.5MM</td>
<td>74%</td>
<td>14.3 months</td>
<td>61%</td>
<td>48%</td>
</tr>
<tr>
<td>2009</td>
<td>$6.2MM</td>
<td>51%</td>
<td>18.4 months</td>
<td>36%</td>
<td>67%</td>
</tr>
</tbody>
</table>

While more than 50% of projects were implemented over budget, the percentage of projects, which took longer than expected went up and it is more than 2/3 of all projects. User satisfaction is lower than in any previous year and this trend is obvious for each year. However, user satisfaction is not always a reliable quality indicator. Decreasing level of user satisfaction could also be a result of organizational issues such as unsuccessful change management based on poorly implemented business model.

“Organizations that take the time to create a business case and define their goals and objectives are well on their way to achieving everything they hope for.” (Panorama Consulting Solutions, 2014)
3. The failure factors

The knowledge of the critical failure factors would allow the project management team to make correct decisions before and during the development of the projects as well as “improve the performance” (Ibrahim, Ayazi, Nasrmalek, & Shamin, May - June, 2013) in project management.

Dr. Roslina Ibrahim et al in their article “An investigation of critical failure factors in information technology projects” have defined four categories of IS/IT project failures:

• “Correspondence failure”. The failure of the incorrect system design.
• “Process failure”. This failure arises if IS projects cannot be implemented within allocated budget or/and time frames.
• “Interaction failure”. That takes place if “the level of end-user usage of the information system is suggested as a surrogate in IS performance measurement” (Ibrahim, Ayazi, Nasrmalek, & Shamin, May - June, 2013). The interaction failure often occurs in a large project where the level of end-users is incorrectly defined as high however “heavy usage does not necessarily mean high user satisfaction and improved task performance”. (Ibrahim, Ayazi, Nasrmalek, & Shamin, May - June, 2013)
• “Expectation failure”. This means the implemented system did not meet customer requirements, expectations or values. (Ibrahim, Ayazi, Nasrmalek, & Shamin, May - June, 2013)

During the research they have also classified five categories of IT project failure by their relations to the other project characteristics:

• Factors related to the project: size & value, uniqueness of project activities, density of a project, life cycle and urgency. (Ibrahim, Ayazi, Nasrmalek, & Shamin, May - June, 2013)
• Factors related to the project manager: ability to delegate authority, ability to trade off, ability to coordinate, perception of his role and responsibilities, competence and commitment. (Ibrahim, Ayazi, Nasrmalek, & Shamin, May - June, 2013)
• Project team members: technical background, communication, troubleshooting and commitment (Ibrahim, Ayazi, Nasrmalek, & Shamin, May - June, 2013)
• Factors related to the organization: top management support, project organizational structure, functional managers' support and project champion. (Ibrahim, Ayazi, Nasrmalek, & Shamin, May - June, 2013)
• Factors related to the environment: political environment, economic environment, social environment, technological environment, nature, client, competitors and subcontractors. (Ibrahim, Ayazi, Nasrmalek, & Shamin, May - June, 2013)

Four common factors of project failure were summarized based on the investigation of a number of researches:

• Poor top management support (Ibrahim, Ayazi, Nasrmalek, & Shamin, May - June, 2013)
• Poor consultant effectiveness (Ibrahim, Ayazi, Nasrmalek, & Shamin, May - June, 2013)
• Poor project management effectiveness (Ibrahim, Ayazi, Nasrmalek, & Shamin, May - June, 2013)
• Lack of user involvement (Ibrahim, Ayazi, Nasrmalek, & Shamin, May - June, 2013)

As we can see, those common project issues can be avoided by improving project management. (Ibrahim, Ayazi, Nasrmalek, & Shamin, May - June, 2013).

Dr. Paul Dorsey in his quite famous “Top 10 reasons why system projects fail” has also indicated top management support as one of the keys to project success. He also described two more keys:

• “Top management support” (Dorsey, 2005)

Management support is a critical success factor in any type of a project, IT or non-IT. Top managers should be prepared and be familiar with the process. Poor management of the project guarantees project failure.

• “A sound methodology” (Dorsey, 2005)

Many software development methodologies were developed over the last 40 years, these include object oriented approach, waterfall, rapid prototyping, etc. The importance is to find the right methodology for the project, however selecting any methodology is always better than non-methodology development process. The selected methodology is not a 100% guarantee of project’s success but at least it gives the expectation for a non-failed result.

• “Solid technical leadership” (Dorsey, 2005)

The technical leader must be familiar with the system that is being planned. For instance if the technical leader is involved in accounting system development he/she should understand basic financial practices. If a technical leader doesn’t really know what he/she is developing, that project is very likely to fail.

Those three keys are a ‘tripod’ in which losing one of the legs will lead to the entire system collapse. (Dorsey, 2005).

If IT specialists, especially managing staff, do not recognize that good engineering principles should be applied to software projects, their projects will possibly fail. The top 10 reasons of failure listed below must be investigated in the early stages of a project if one of them takes place. “The following list has been inspired by actual mistakes encountered in real-world system projects” (Dorsey, 2005):

• Using unspecified methodology (Dorsey, 2005)
• Project planning from a drop-dead system completion date. (Dorsey, 2005)
• Non using a Data model (Dorsey, 2005)
• Using a technical leader without experience in similar projects (Dorsey, 2005)
• Using more developers to accelerate a project (Dorsey, 2005)
• Using programming language which developing team never used before (Dorsey, 2005)
• Using inexperienced developers for data migration (Dorsey, 2005)
• Skipping testing phases as unimportant (Dorsey, 2005)
• Fulfilling the critical new requirements discovered during final implementation by changing the system (Dorsey, 2005)
• Customizing a commercial “off-the-shelf” package (Dorsey, 2005)

In contrast to those listed reasons Dr. Paul Dorsey described most important factors for system success:

• Methodology - the cornerstone of project planning
• “Audit each major deliverable and step along the way for accuracy and correctness.” (Dorsey, 2005)
• The obligation to support the project by top management
• Securing “the correct technical leader for the project” (Dorsey, 2005)

In the report published by Oracle Corporation “Why Projects Fail: Avoiding the Classic Pitfalls” six areas are mentioned as responsible for most common IS project failures:

Constituent Alignment. To be successful the projects should be clearly defined in terms of their goals and objectives. “Any course corrections or even project cancellations” should be part of regular project management. Remember, it takes the average person seven views of the same message before it starts to sink in. (Oracle Corporation, 2011)

Proactive Risk Management. Risk management is often a missing factor of project failure. To achieve the success in project management it is absolutely necessary to identify risks and integrate them to “scope, schedule and cost.” (Oracle Corporation, 2011)

Performance Measurement. Correct measurement of project is one of the most important factors of project success rate. Only with correct measurement the key persons can recognize current or appearing issues. As a result, “troubled projects are not highlighted in time for remedial action”. “A recommended approach is to use standardized project performance measures”. Even in small projects Earned Value Management (EVM) should take place. (Oracle Corporation, 2011)

Project Scope Definition and Management. The project scope “must be clear, concise, and unambiguous”. All parties of the project including stakeholders, team members etc. should clearly understand project definition, requirements, and other information which is normally included in agreement and later in software development project. “A clear understanding of scope is essential to gaining commitment and executing successfully”. (Oracle Corporation, 2011)

Critical Project Communication. Any part of project can be affected by this issue. Project managers and stakeholders should be informed about issues on time and before the impact on costs, timelines and scope become “significant or irreversible”. To be aware of this pitfall the communication management plan should be created and “comprised of two parts: project communications and stakeholder communications”. (Oracle Corporation, 2011)

Methodology Usage. The importance of methodology is “often overlooked”. A number of project management approaches is available and standardized; however, some companies can use more specific or internally developed methodologies. The right choice and enforcement of methodology is a critical factor for project management, employing either standardized methodology or organization-specific. (Oracle Corporation, 2011)
The projects “do and will fail” (Oracle Corporation, 2011) and the IS project is not an exception. The knowledge of described factors and pitfalls “will not guarantee a successful project” (Oracle Corporation, 2011). However, it provides “a solid footing and foundation from which to begin the process of executing against the project’s objectives and strongly influence a successful outcome”. (Oracle Corporation, 2011)

In addition, most common failure causes can be classified as well. Calleam Consulting Ltd. in their information web-site has pictured 8 primary categories of those causes.

Market and strategy failures – Before building any product the specific investigation should indicate that this is the right product as there are some users in market who require this product.

Organizational and planning failures – The project should be properly and effectively organized. “Where the level of organization is insufficient the project team can quickly lose control.” (Calleam Consulting Ltd., 2014) However, in some projects the control has been “inappropriate for the type of project being run” (Calleam Consulting Ltd., 2014) and as a result “the project can be weighed down by unnecessary inefficiencies”. (Calleam Consulting Ltd., 2014)

Leadership and governance failures – Ineffective leadership and low level of governance processes: management reduce the control of a project or even create the conditions in which a project cannot be continued.
Underestimation and analysis failures – Project complexity must be correctly understood at the analysis phase and “before commitments to schedule and budget are confirmed”. (Calleam Consulting Ltd., 2014) Unrealistic estimation was the reason for failure in a number of large government projects.

Quality failures – “Sadly quality is often the dimension that gets too little attention”. (Calleam Consulting Ltd., 2014) In the projects where the management staff closed their eyes on truncated quality corners or insufficient testing results, “serious flaws can escape into the project and cause havoc once the deliverables have been deployed”. (Calleam Consulting Ltd., 2014)

Risk failures – Where project managers are “blind” to the possibilities of risks, such projects are “likely to run into serious difficulties that they failed to anticipate”. (Calleam Consulting Ltd., 2014)

Skills, knowledge and competency failures – Inadequate skills, knowledge and competency of people involved in the project “increase the chance of project success”. (Calleam Consulting Ltd., 2014)

Engagement, teamwork and communications failures – The project “is working in a vacuum” (Calleam Consulting Ltd., 2014) if there are insufficient communication between project players such as stakeholders etc. The same issue happens if the team members ineffectively collaborate between them.

It is interesting to investigate the failures in relation to project size. It was found in Gartner user survey (Gartner, Inc. is the world’s leading information technology research and advisory company) done at the end of 2011 that “half of all project failures, irrespective of project size, were put down to functionality issues and substantial delays”. (Mieritz, 2012)

The analysis below shows the result for a five-country survey. Here we can see the relationship between the failure rate of IT projects and their budgets. This is clearly an indication that “small IT projects experience a one-third lower failure rate than large projects”.

![Graph showing percentage of respondents by project size and success/failure rates](image-url)
During the survey the respondents were asked to mention reasons or causes for project failure over the past two years “across six frequently mentioned reasons or causes of project failure”:

- Functionality issues
- Substantially late
- Quality issues
- High cost variance
- Cancelled after launch
- Rejected or not implemented for other reasons

The result was condensed into the graph form with distribution by project size:

Based on this graph we can define some recommendations for project management:

- The success is more likely if the project size is limited;
- Project managers (especially for the larger scale projects) should ensure that “there are appropriate mechanisms in place to identify budget variances and/or overruns early” (Mieritz, 2012)
- The schedule has to be realistic. That is more important for large projects because “business conditions keep changing after the project scope has been set, leaving a significant disconnect between the agreed-on scope and budget versus what the business will require and pay for by the time the project is delivered.” (Mieritz, 2012)
- Project managers must increase the “frequency of project status and review meetings, as well as ongoing confirmation of the project’s alignment with business strategy”. (Mieritz, 2012) It will give ability to identify issues and/or cancel projects at the earliest possible stage.

The criteria for failure can be stipulated by three categories: functionality, lateness, and cost overrun. Other factors that can be included in the definition of failure are poor quality work and cancellation of the project after launch.
The failures of IT project. Factors and early prediction

The reasons for project failure based on the 2012 Gartner Survey are:

- “Unrealistic schedule” (O’Brien, 2013) and reactive management which is more about project manager skills and experience in similar projects.
- “Shifting requirements” (O’Brien, 2013) currently is a very common issue originating from poor initial requirements and other BA jobs.
- Ineffective teamwork, e.g “insufficient skill sets”. (O’Brien, 2013) The team should be able to reach the targets of the project. Hence the team should be selected carefully, ensuring that team members have sufficient experience in similar projects and clear understanding of the required establishment strength.

As we see, the critical situation is typical for larger scale projects, while small projects overall have less percentage of failure.

Below are two charts (based on the 2012 report) where the small projects are the projects with the less than $1 million budget in labour costs and the large projects - more than $10 million in labour costs.

![Pie charts showing failure rates for small and large projects.](image)

(The Standish Group, 2013)

Thus the practice of breaking down large projects into smaller ones gives the opportunity to significantly increase the percentage of successful projects. There are many methodologies that work in this kind of project environment. However it also requires specific style of project management.

Chaos group in its research of 2013 has identified factors of success for small projects. The picture is quite different to the large scale projects.

**Executive Management Support.** The executive sponsor is the person who is responsible for the whole project, for its success and failure. "Small projects allow for a new or inexperienced executive sponsor”. Having a skilled and engaged executive is a good option however it could be a “midlevel manager—or in the case of agile, a product owner”. (The Standish Group, 2013)

**User Involvement.** Obviously in large projects skills and responsibility of users and the project team significantly effect the project outcome. However “small projects allow a less
strict or burdensome approach in dealing with users and users groups”. (The Standish Group, 2013)

**Optimization.** Smaller size of a project has more probability of success. Most of large projects can be broken down into small projects. We must not forget that a small project can be risky too but this is often “because they generally are focused on accomplishing something new and unknown”. (The Standish Group, 2013)

**Skilled Resources.** Easily defined as “having the right people doing the right things at the right time.” (The Standish Group, 2013)

**Project Management Expertise.** “Experienced, seasoned, and talented project managers can handle multiple small projects”. (The Standish Group, 2013) However it is not the best practice. Better to merge some functions for one manager in a small team. With agile methodology process master handles a number of functions and one of these functions is project management.

**Agile Process.** : “The agile process benefits from small projects”. (The Standish Group, 2013) Small projects have a better success rate. In this case agile methodology makes it easier to execute small projects and/or “break up larger projects into a series of smaller projects”.

**Execution.** “Execution is the act of taking a project to completion based upon a plan.” (The Standish Group, 2013) Small project are easier accomplished than large ones.

**Tools and Infrastructure.** Small organizations that are working on small projects are limited in available tools and infrastructure to increase education, competency etc. However using cloudbased services is one of many ways that helps “move projects forward without committing fully to one direction.”

One of the best answers for the question ‘how to improve the percentage of successful IS projects?’ is “to reduce the size and complexity” including “starting all projects small or breaking up large projects into small projects”.

It is impossible to “eliminate all project failures”, (O’Brien, 2013) however a good project manager can and has to know how to detect failure in its very early stage, before those issues will affect the whole project.
4. Early prediction of failure

The obvious fact that the earlier the fault will be found the less will it cost to fix is self-evident.

The factors contributing to the success or failure of IS projects divided by 7 categories (Procaccino, Verner, & Overmyer, 2000):

- requirements
- management
- customers and users
- estimation and scheduling
- the project manager
- the software development process
- development personnel.

In their research “Case Study: Factors for Early Prediction of Software Success & Failure” J. Procacciono et al had investigated 42 quite large IS projects using questionnaires where the questions were also separated by the categories listed above. Later the answers were analysed.

The questionnaires have been addressed to the managers and developers to see the differences in their points of view. It was interesting that “overall, while management thought 78% of our projects were successful the respondents considered only 49% of the same projects a success.” (Procaccino, Verner, & Overmyer, 2000)

The summary of analysis was divided in three groups by analytical category. Knowledge of these outcomes helps to predict possible failures:

Management support category

- “A project can be considered successful even when a project manager is not given full authority to manage the project.”
- “Starting a project with a committed sponsor is important to development staff, but not necessarily to management.”
- “Having a project sponsor who is visibly committed for the duration of a software project is key to project success from both management and developer view.”
- “A project sponsor who drops out of their role in the project has a more detrimental effect on perceptions of project success than starting without a sponsor and picking up one later”
- “Projects can be perceived as successful without the sponsor’s involvement in project decision-making. This helps define the role of project sponsor as political, rather than production support.”
- “If other stakeholders are both committed and involved with a project, the developers will feel better about it but not significantly so. This commitment appears to be irrelevant to managers.”

(Procaccino, Verner, & Overmyer, 2000)
Customer/users category

- “If customers/users are involved to at least a reasonable level, a project is more likely to be viewed as a success by management and developers.”
- “Both developers and management will be more likely to view the project as a success if the customers/and users stay involved right throughout the project.”
- “The higher the level of confidence that customers/users have in the project manager and development team the more likely developers are to consider the project a success. Managers’ perceptions of project success do not seem to be impacted by stakeholder confidence.”
- “The involvement of customers/users in making schedule estimates does not increase the likelihood that developers will consider the project a success, however, their input appears to be important to managers.”
- “Low customer turnover appears to increase the perception of project success by developers, but not managers. This is easily explained by observing that it is the developers who must contend with customer/user turnover, not the managers.”
- “Developers are more likely to consider a project a success if the customers/users have realistic expectations. Customer/user expectations also appear to make a difference in management perceptions of project success, but not significantly so.”
- “Large numbers of customers/users did not affect anyone’s perception of the success of these projects. Having large user populations can be a double-edged sword, creating as much conflict as consensus.”

(Procaccino, Verner, & Overmyer, 2000)

Requirements category

- “If requirements are not compete and accurate at the start of a project developers are less likely to view that project as a success. The lack of accuracy and completeness of requirements, at the start of a project does make any difference to management’s perception of project success.”
- “If requirements are not completed adequately during the project neither management nor developers are likely to consider the project a success.”
- “Using a requirements gathering methodology did not appear to increase the perception of project success by either developers or management.”
- “If the scope of a project is well defined, management and developers are more likely to view the project as a success”
- “Changing the scope of a project during development does not change either management’s or developers’ perception of project success.”
- “A central repository for requirements was important for success for developers but made no difference to management’s view of project success.”
- “When customers/users do not make adequate time available for project requirements gathering developers are likely to consider the project less successful. It does not change management’s perception of the success of projects if customers and users do not make adequate time available for requirements gathering”
- “Well defined software deliverables were important for the success of a project in developers view but did not matter to management”
“Projects so large that the requirements elicitation is impacted were likely to be less successful so far as developers were concerned but this did not matter to management”

(Procaccino, Verner, & Overmyer, 2000)

Conclusively, this comprehensive research gives some fresh ideas for the management staff:

1. The perceptions of people involved in the project “can dramatically affect the health of a project manager, a project team, and thus the health of the project.” (Procaccino, Verner, & Overmyer, 2000)

2. A vast difference has been found "between developers’ perceptions of project success factors, and their perceptions of how management personnel view project success and failure” (Procaccino, Verner, & Overmyer, 2000)

From the developers’ point of view, a successful project (and project manager) should have:

- customers/users who make adequate time available for them (a project sponsor will help here)
- a project scope that is well-defined
- carefully managed customer/user expectations

The management viewpoint “appears to be more politically oriented - towards keeping customers and users (rather than developers) happy, and passing (a share of) the blame to someone else if the schedule is not met”. (Procaccino, Verner, & Overmyer, 2000) Current level of IT management needs to keep developers happy and protect the staff from turnover and loss of motivation.

The cancellation of a project in time is a painful but an essential skill for a project manager. Even if the project has already started the way can and must still be found to cancel the project before it will be too late to protect the main part of investment.

Potential failure of the project is indicated by certain symptoms visible to the project manager and they should become a reason for an investigation:

- “frequent requests by users to change the system” (McManus & Wood-Harper, 2008)
- insufficient communication between development team and stakeholders
- unclear definition of requirements

Those symptoms would be an early indicator of some issues in business process alignment, requirements management or of possible overspending.

Key reasons for cancellation of a project have been collected in the article written by Dr. John McManus and Dr. Trevor Wood-Harper in their “A study of project failure”: 
Business reasons for project failure

- “Business strategy superseded”;
- “Business processes change (poor alignment)”;
- “Poor requirements management”;
- “Business benefits not clearly communicated or overstated”;
- “Failure of parent company to deliver”;
- “Governance issues within the contract”;
- “Higher cost of capital”;
- “Inability to provide investment capital”;
- “Inappropriate disaster recovery”;
- “Misuse of financial resources”;
- “Overspends in excess of agreed budgets”;
- “Poor project board composition”;
- “Take-over of client firm”;
- “Too big a project portfolio”.

(McManus & Wood-Harper, 2008)

Management reasons

- “Ability to adapt to new resource combinations”;
- “Differences between management and client”;
- “Insufficient risk management”;
- “Insufficient end-user management”;
- “Insufficient domain knowledge”;
- “Insufficient software metrics”;
- “Insufficient training of users”;
- “Inappropriate procedures and routines”;
- “Lack of management judgement”;
- “Lack of software development metrics”;
- “Loss of key personnel”;
- “Managing legacy replacement”;

(McManus & Wood-Harper, 2008)

Poor vendor management

- “Poor software productivity”;
- “Poor communication between stakeholders”;
- “Poor contract management”;
- “Poor financial management”;
- “Project management capability”;
- “Poor delegation and decision making”;
- “Unfilled promises to users and other stakeholders”.

(McManus & Wood-Harper, 2008)
Technical reasons

- “Inappropriate architecture”;
- “Insufficient reuse of existing technical objects”;
- “Inappropriate testing tools”;
- “Inappropriate coding language”;
- “Inappropriate technical methodologies”;
- “Lack of formal technical standards”;
- “Lack of technical innovation (obsolescence)”;
- “Misstatement of technical risk”;
- “Obsolescence of technology”;
- “Poor interface specifications”;
- “Poor quality code”;
- “Poor systems testing”;
- “Poor data migration”;
- “Poor systems integration”;
- “Poor configuration management”;
- “Poor change management procedures”;
- “Poor technical judgement”.

(McManus & Wood-Harper, 2008)

There is no doubt that large national projects are more likely to fail. Moreover, the larger the project the more factors need to be considered to predict the failures and to avoid them. E-government projects can be good examples of national projected investment by governments. Based on the report by Carolyne Stanforth “Analysing e-Government Project Failure: Comparing Factoral, Systems and Interpretive Approaches” we can learn how to diagnose failures in the large national projects. The report has shown some more factors that haven’t been mentioned earlier.

Carolyne Stanforth has reviewed “the three main categories of diagnostic approach being used in the study of failed e-government projects – factoral analyses, systems approaches and interpretive studies.” (Stanforth, 2010)

Factoral approaches. “The essence of the approach is that failure of a project to meet its targeted performance is caused by a combination of these identified critical factors”. (Stanforth, 2010). In this approach the acronym ITPOSMO is a summary of seven dimensions of potential design-reality gaps which need to be investigated:

- Information
- Technology
- Processes
- Objectives and values
- Staffing and skills
- Management systems and structures
The failures of IT projects. Factors and early prediction

- Other resources (time, money etc.)

The ITPOSMO [Information, Technology, Processes, Objectives, Skills, Management Systems, Other Resources] model allows to identify three “archetypes of failure” (Stanforth, 2010):

- Hard-soft Gaps (the difference between technology design and socio-political reality in which system should operate)
- Private-Public Gaps (the difference between a system developed for private sector and its usage in the public sector)
- Country-Context Gaps (the difference between a system developed for one country and used in a different country).

System approaches. Systems approaches are based on the study of disasters, or, in the other words, of catastrophic types of system failures. In this approach the information system should be validated as “being the fit between the system design attributes and organization realities on four levels” (Stanforth, 2010):

- User-System level (between the psychological characteristics of users and the system design)
- Structure-System level (between the structure of organization and the system design)
- Power-System Level (“between the distribution of power in the organization and the system design”)  
- Environment-System Level (between the organization environment and system design).

Interpretive approaches. This approach comes from the social sciences. “They are based on the belief that 'the system' is the construction of an individual, often established with the aim of interpreting it or understanding it more fully.” (Stanforth, 2010)

One of the most interesting frameworks for this approach has been created by Davis et al (1992) as a two-dimensional matrix for diagnosing information system failure.

The matrix displays four areas of "potentially useful data for a failure diagnosis" (Stanforth, 2010) on each dimension.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Social System</th>
<th>Technical System</th>
<th>Reactions to technical system</th>
<th>Performances indicators</th>
<th>Development processes</th>
<th>Designers' theories-in-use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technology</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>User interfaces</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Information requirements</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Organisational fit</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

Socio-Technical Framework for Data Analysis (Davis et al, 1992)

The data in the matrix is classified using the sixteen cells. Any apparent anomalies are the points for further investigation.
The investigation process works as a loop in which information about project is gathered and the analyst looks for apparent discrepancies, absurdities or discontinuities. Next the analyst “seeks further information to provide an answer to each disconnect, then looks again at the situation and develops a new understanding of it” (Stanforth, 2010) until the whole picture will be revealed.

Nearly all large projects include software development as their main part. Software development process has to be systematically detected against the faults/issues using special techniques from the workflow requirements to the logical end of development. Quality assurance team discovers faults in the started project plans.

Statistically “between 60 and 70 percent of all faults detected in large projects are requirements, analysis, or design faults.” (Schach, 2011) That obliges the project manager to employ some “techniques for detecting faults” (Schach, 2011) at all stages of SDLC. Testing software just once before the delivery to the client is too late and not sufficient. In this case continuous testing of the software product must be organized.

It means that testing is “an integral component of the software process ad an activity that must be carried out throughout” (Schach, 2011) during the entire life cycle of the project. “During the requirements workflow, the requirements must be checked; during the analysis workflow, the specification must be checked; and the software production management plan must undergo similar scrutiny”. (Schach, 2011) Based on IEEE610.12, the failure is “the observed incorrect behaviour of the software product as a consequence of a fault, and the error is the amount by which a result is incorrect”. (Schach, 2011) And also “a fault is injected into the software when a human makes mistake”. (Schach, 2011)

Before the beginning of software coding there is only one way to ensure the quality of software, and that is non-execution-based testing. During this process the QA team reviews the project documentation in two ways: walkthroughs and inspections. The reviews of documentation is “an effective way to detect a fault” (Schach, 2011) at the early stages of development. As a result this fault will be inexpensive to fix - or the project can even be cancelled with less stress and investment if the number of discovered faults is significant. The project management staff should ensure the effectiveness of review process especially in large-scale projects where any testing activities are extremely hard unless the project is broken down into independent smaller components, or, in other words, unless we utilize an object-oriented paradigm.

Some reasons of IT project failure are obvious and some are not well recognised. However, PMs should bear responsibility to identify all those possible factors at their earliest stages. Below are the ‘symptoms’ of already ‘sick’ projects:

“Fuzzy goals” (Kogekar, 2013) – there is a complex of issues when the possible project achievement cannot be clearly defined. This reason includes unclear goals, poor project design and other project mistakes that makes project mission blurred.

“Over-optimism” – this a current tendency to maximize the benefits and reduce the cost of project at the same time. That is a result of poor project managers’ understanding of the requirements and current project situation as a whole.
“Complexity” – high degree of complexity in current IT project became normal. However, project managers do not always understand that “the project risks and effort involved increase exponentially as complexity increases”. (Kogekar, 2013)

“Weak ‘ownership’” – More than one involved project manager in the large project is normal practice. As a result “none of the executives fully support the project”. (Kogekar, 2013) Moreover the conflicts between them are very possible. It also indicates “poor accountability across the project” and poor project control. (Kogekar, 2013)

“Governance” – obviously lack of governance is a major reason for the failure of any project. However very large IT projects display another side of this problem which is too much bureaucracy, slowing down documentation moving and decision making processes.

“Over-engineered thinking” – project managers become more focused on modern planning techniques than on the mission itself. They spend time and resources “planning the methodology and approach to the project rather than working on the technical requirements of the solution and how we can deliver it.” (Kogekar, 2013) Furthermore the numbers and volume of documentation are increasing every day which also causes more bureaucracy and more non-development staff.

“The management factor” – the project manager is a factor him/herself. Certified and experienced PM is not everything. “The PM has to understand how to balance the needs of the project with the needs of good governance. The PM has to use the right methodology, apply it at the right time and in the amounts that are really needed.” (Kogekar, 2013)

“Think big, act small” – smaller projects have higher success rate than larger projects. So the large project should be broken down into small stages and become “a group of small projects” (Kogekar, 2013). That significantly decreases the total complexity and possibly gives the ability to work with less number of personnel (especially non-development staff).
5. Faults in real projects

The information on real projects has collected by reviewing official reports and provided in support of the research.

2005. THE FEDERAL BUREAU OF INVESTIGATION’S MANAGEMENT OF THE TRILOGY INFORMATION TECHNOLOGY MODERNIZATION PROJECT

Quite famous failed project had been developed for FBI between 2001 and 2009. The Trilogy project was planned as an upgrade of the FBI’s previous IS. First contracts were signed in 2001 with two companies as was required because the project was considered to be too large for one developer. Later the requirements, developers and methodology were changed.

Factors below have been identified in the report of US Department of justice (US Department of Justice Office of Inspector General Audit Division, 2005):

- poorly defined and evolving design requirements
- contracting weaknesses,
- IT investment management weaknesses,
- lack of an Enterprise Architecture,
- lack of management continuity and oversight,
- unrealistic scheduling of tasks,
- lack of adequate project integration, and
- inadequate resolution of issues raised in reports on Trilogy.

Those factors resulted in delays and significant over-budgeting of the project.

Poorly Defined and Slowly Evolving Design Requirements. “Trilogy’s design requirements were ill-defined and evolving as the project progressed. In addition, certain events (11/09/2011 etc.) required the need to modify initial design concepts.” (US Department of Justice Office of Inspector General Audit Division, 2005)

Contracting Weaknesses. The initial contacts did not require specific completion milestones, did not include critical decision review points, and did not provide for penalties if the milestones were not met. “Contracts are only required to make their best effort to complete the project.” (US Department of Justice Office of Inspector General Audit Division, 2005)

IT Investment Management Weaknesses. “The IT Investment Management (ITIM) process was not well-developed”. (US Department of Justice Office of Inspector General Audit Division, 2005) ITIM process was “missed milestones and uncertainties associated with Trilogy”. (US Department of Justice Office of Inspector General Audit Division, 2005) “The development had been managed in a “stovepipe,” and as a result FBI personnel not involved in the management, had little knowledge of the project’s status and progress.” (US Department of Justice Office of Inspector General Audit Division, 2005)

Lack of Enterprise Architecture. “FBI did not have an Enterprise Architecture, although it began developing one in early 2000” (US Department of Justice Office of Inspector General Audit Division, 2005); also FBI did not have the management structures and processes to effectively develop.
Lack of Management Continuity and Oversight. “Since November 2001, 15 different key IT managers have been involved with the Trilogy project”. (US Department of Justice Office of Inspector General Audit Division, 2005)

Unrealistic Scheduling of Tasks. “Along with the lack of firm milestones in the Trilogy contracts, the scheduled completion dates for individual project components were unrealistic.” (US Department of Justice Office of Inspector General Audit Division, 2005)

Lack of Adequate Project Integration. “Despite the use of two contractors to provide three major project components, the FBI did not hire a professional project integrator to manage contractor interfaces and take responsibility for the overall integrity of the final product until the end of 2003.” (US Department of Justice Office of Inspector General Audit Division, 2005)

Inadequate Resolution of Issues Raised in Reports. “The FBI’s management of its IT, including the Trilogy project, was the focus of several reports issued both by components within the FBI and external reviewing entities”. (US Department of Justice Office of Inspector General Audit Division, 2005) The report demonstrated that the “FBI took inadequate actions to resolve the findings of the reports”. “Because of the lack of resolution, many of these issues have remained throughout the course of the project.” (US Department of Justice Office of Inspector General Audit Division, 2005)

2011. THE BBC’S MANAGEMENT OF ITS DIGITAL MEDIA INITIATIVE

The BBC’s Digital Media Initiative is the IS which is “designed to allow BBC staff and partners to develop, create, share and manage video and audio content and programming on their desktop”. (Committee, Controller and Auditor General presented to the BBC Trust’s Finance and Compliance, 2011)

The report made for BBC Trust identified some key points that affected the project:

1. “The BBC did not run an open procurement competition before awarding to Siemens the contract to deliver the Programme.” (Committee, Controller and Auditor General presented to the BBC Trust’s Finance and Compliance, 2011) The market and competition haven’t been investigated. The contract has been signed with Siemens as with company which already worked for BBC. “As a result the BBC did not have as strong assurance on price, quality and capacity to deliver as a new and specific tender may have provided.” (Committee, Controller and Auditor General presented to the BBC Trust’s Finance and Compliance, 2011)

2. The contract was terminated “by mutual agreement with effect from the end of July 2009.” (Committee, Controller and Auditor General presented to the BBC Trust’s Finance and Compliance, 2011)

3. “The BBC took responsibility for delivery of the Programme in July 2009, accepting the risks of delivering the Programme, although without testing the value for money of this approach.” (Committee, Controller and Auditor General presented to the BBC Trust’s Finance and Compliance, 2011) BBC didn’t try to examine any other option such as finding a new contractor.
4. “The BBC’s in-house delivery of the system has started well”. (Committee, Controller and Auditor General presented to the BBC Trust’s Finance and Compliance, 2011) However, after delivering two first components, the project was delayed again and delivery plan was revised “because of delays in defining procurement requirements”. (Committee, Controller and Auditor General presented to the BBC Trust’s Finance and Compliance, 2011)

5. “The financial case for the Programme has weakened over time” (Committee, Controller and Auditor General presented to the BBC Trust’s Finance and Compliance, 2011)

6. “There was a marked improvement in the June 2010 investment case approved by the BBC Trust compared with earlier cases.” (Committee, Controller and Auditor General presented to the BBC Trust’s Finance and Compliance, 2011)

A number of factors could be used as indicators of possible failure. However, they were missed and/or wrongly interpreted:

- competitive tender has not been organized. However it is legally recommended for the high value projects. “Without transparently testing and explicitly comparing the costs and benefits of other options for delivering the Programme, the Programme team was unable to demonstrate to the Trust that the chosen delivery course optimised value for money.” (Committee, Controller and Auditor General presented to the BBC Trust’s Finance and Compliance, 2011)

- Defined 30 days elaboration phase hasn’t been completed on agreed time (March 2008). Elaboration phase was completed in June 2008 - as a result the design phase started also later than agreed. “In fact the two parties did not reach agreement on the detailed design for the system and the Programme never reached the User Acceptance Testing phase.” (Committee, Controller and Auditor General presented to the BBC Trust’s Finance and Compliance, 2011)

Two key milestones in November 2008 and in March 2009 were defined, however first key milestone in November 2008 was likely to be delayed by three months.

- After the termination of contract with Siemens in-house development was started. However the assessment of BBC’s capacity and capability hasn’t been done. It was understood that “the plan was prepared without full understanding of the technical and design issues that Siemens had encountered and that as it did not have the capacity or capability to deliver the system it would have to recruit or use a number of third party suppliers to fill this gap”. (US Department of Justice Office of Inspector General Audit Division, 2005)

- It was correctly concluded “that an approach with ‘big bang’ deliveries was not appropriate for this type of programme involving the development, procurement and integration of different technologies.” (Committee, Controller and Auditor General presented to the BBC Trust’s Finance and Compliance, 2011) Development methodology has been reconsidered to a more iterative ‘agile’ approach. The development process has been broken down into “smaller steps, with quick feedback loops from users to improve the products being developed”. (Committee, Controller and Auditor General presented to the BBC Trust’s Finance and Compliance, 2011)

- Due to the selected agile methodology “the BBC has well documented processes for integration testing and user acceptance testing and has carried out such testing successfully on
2013. 6 SOFTWARE DEVELOPMENT LESSONS FROM HEALTHCARE.GOV’ FAILED LAUNCH

Healthcare.gov is one of the most famous project failures in the past 10 years. The article “6 Software Development Lessons From Healthcare.gov's Failed Launch” written by Matthew Heusser is more about the lessons we can learn from that project than about just listing collections of obvious facts of failure.

Methodology issue. Management staff probably hasn’t fully understood what the main difference between Agile and Waterfall methods is. Development team was using ‘sprints’ or ‘iteration’ into wrong concept which was more waterfall than agile. In agile approach “the terms "sprint" and "iteration" mean the time for a new, completed chunk of software development to be designed, coded and fully tested, end-to-end”. (Heusser, 2013) However in this project sprints were separated, as in "three architecture sprints, six coding sprints, two test sprints and two hardening sprints". (Heusser, 2013) We can call it ‘sprint’ however it does not mean that it is fundamental agile concept.

Wrong statement employed: If system produces outcome it doesn’t need to be tested. System wasn’t fully tested. Even if it works it should be tested and the testing phase should be defined as a legitimate phase in planned development approach. “The problem at Healthcare.gov wasn’t lack of testing. It was the lack of critical thinking.” (Heusser, 2013)

“Testing Should Be Part of the Delivery Process” (Heusser, 2013). The entire process is firefighting. Testing wasn’t seriously considered. If testing actually occurred — and real testing, not fake testing — it failed to make any meaningful connection to operations or support or project management. Testing failed to recommend a no-go decision.

“Do It Manually Before You Automate” (Heusser, 2013). This recommendation means that developers should keep the possible way to use previously used manual processing in case launching process would fail for some reason.

“The System Had to Be Perfect, By Law, But It Wasn't” (Heusser, 2013). This is a requirements issue. Users were unable to use the system until they registered. “Healthcare.gov must verify a would-be applicant's eligibility from Homeland Security, IRS and Social Security systems, in real time, plus enrol members electronically to any registered insurance company in any of the 34 states that don't have their own state health insurance exchange.” (Heusser, 2013) “Information needs to be completely accurate, so every request needs to run - submit information, over Web services, to an insurance company. That means users click Submit, then wait for a back-end Web services call, and wait and wait and ...” (Heusser, 2013) Healthcare.gov works in the way opposite to normal practice. It forces to submit personal information before allowing users to see a quote. “That allowed Healthcare.gov to force users to agree to terms of service, and to connect to back-end servers at the IRS, Homeland Security and other agencies. Those are great features for the end of the process, but when your users want to window shop, perhaps less so.” (Heusser, 2013)

Modelling issue. The system can be hacked even without special knowledge. Investigation of the source code with the knowledge of how to use it allows to hack the system. For
instance, “reset feature results in data combinations that could enable phishing attacks. An attacker logged in as you can get personal information from RESET service responses, including name, address, date of birth and Social Security number.” (Heusser, 2013) “It’s likely that security testing was faked, overlooked or ignored for Healthcare.gov, just like functional testing. Don’t let this happen on your watch.” (Heusser, 2013)

2013. NOVOPAY TECHNICAL REVIEW. FINAL REPORT

Novopay is an example of initially failed project implemented despite the ‘sensus communis’, with the large over-budgeting and delaying.

The “Report of the Ministerial Inquiry into the “Novopay Technical Review” defined key issues in the process of project implementation after nearly 3 years since the date when this project should have been implemented.

1. “In some areas system functionality does not adequately support the business processes.” (Deloitte. A member of Deloitte Touche Tohmatsu Limited, 2013) There can be only one conclusion - that the business process was not investigated in detail and the requirements for the system were poorly defined from the start.

2. “Usability issues and lack of data input validations contribute to processing errors.” (Deloitte. A member of Deloitte Touche Tohmatsu Limited, 2013) This is an adoption issue. A large system needs more adaptation or/and modification if the developers employed previously implemented ‘as-sold’ product.

3. “School management visibility and control is limited by reports that are sometimes poorly presented or inconsistent.” (Deloitte. A member of Deloitte Touche Tohmatsu Limited, 2013) This type of issue is a result of poor requirements. Developer’s BA staff should gather information not only from the customer (who has no planning as an End User). So the end user needs were not in focus of the project.

4. “Data quality has been affected by system issues, raising the risk of future errors”. (Deloitte. A member of Deloitte Touche Tohmatsu Limited, 2013) Because the initial data has been converted from the previous system which worked well, there is only one conclusion - that the conversion has been made incorrectly and has not been fully tested afterwards either.

5. “Quality controls of data entry have not adequately prevented errors.” (Deloitte. A member of Deloitte Touche Tohmatsu Limited, 2013) That is clearly a mistake of the Interface testing team because the interface itself should prevent as much incorrect information from end users as possible.

6. “A high degree of customisation in high-impact areas has made on-going development more difficult.” (Deloitte. A member of Deloitte Touche Tohmatsu Limited, 2013) The very broad customisation of ‘as-sold’ system created a mass of mistakes.

7. “Aspects of the application architecture make customisation difficult.” (Deloitte. A member of Deloitte Touche Tohmatsu Limited, 2013) Ministry of Education hasn’t received full documentation of the solution architecture. The risks of future difficulties in the support and upgrade of the system were thus clearly defined.
8. Service support processes have struggled to resolve the volume of issues. The developer does not have and has not prepared enough support staff and did not provide requested formal process documentation either. The management process was defective as it has not been organized in a professional manner.

Each of those issues could have been prevented in the early stages by the right management. So issues 1, 3 and 7 are the results of ineffective analysis. Issues 2, 4 and 6 are related to the data and adoption quality and might be grouped as the issues of poor requirement process. Issue 5 is a specific issue of inadequate quality management process. And the last issue says that the management of post-delivery process was also organized inaccurately, although for the systems where the number of end-users is about 100,000 the launching process and maintenance phase have to be well planned and organized long before they begin.
6. Conclusion

As we could clearly see, based on collected researches and case studies, the factors of IS project failure are repeated from project to project. Some of these factors can be recognised as common project management issues. However, some of the defined factors are specific for IS projecting. And moreover, software development as a part of IS projects has more specific failure factors.

Project management is a well-known branch of studies with long history, starting from the building of Egyptian Pyramids, which totally changed after the First Industrial Revolution, assumed more modern forms after the Great Depression and finally developed in 60s of the XX century. On the other hand, IS project management is a very new science that was born at a time when the corporate economy sector decided to buy more than one computer, only 20 years ago...

Thus there is a historically short time period which was not enough for forming strong discipline and standards, such as exist in building project management, manufacturing management etc. Owing to modern communications, which is also the result of the new IS technology, we are able to collect reports, researches, case studies and classify found factors into a tree structure.
The failures of IT project

Factors and early prediction

The tree diagram which we introduced above reveals the number of factors which actually determine the project success. Project manager or project management team (if it is a large project) must systematically monitor the situation and make prompt decisions to steer the process in the right direction. The decision to stop or cancel the project is also possible even if the project has already started, because in some conditions early cancellation can save investment better than production of unsuccessful product/project finally.

Below is the table with factors that can be detected as potential failure reasons. All factors were defined from the diagram.

The failures of IT projects. Factors and early prediction 30
<table>
<thead>
<tr>
<th>factors</th>
<th>description</th>
<th>affection</th>
</tr>
</thead>
<tbody>
<tr>
<td>System doesn’t meet requirements</td>
<td>Requirements were over sighted in the initial phase, or formulated with a</td>
<td>Low client satisfaction or complete fail as product unusable</td>
</tr>
<tr>
<td></td>
<td>number of mistakes. Requirements do not relate to the real client needs.</td>
<td></td>
</tr>
<tr>
<td>Incorrect system design</td>
<td>The mistakes in analyst and architecture jobs which can be defined by an</td>
<td>Project will fail or will be significantly over-budgeted because the team</td>
</tr>
<tr>
<td></td>
<td>experienced PM at the earliest stage of the implementation process.</td>
<td>will need to return to the analysis stage.</td>
</tr>
<tr>
<td>Incorrectly defined end-user</td>
<td>End-user level should be defined before the project starts because the user</td>
<td>Project will be affected at the final stage which is more painful for the</td>
</tr>
<tr>
<td>level</td>
<td>involvement and level of preparation should be included in the project plan.</td>
<td>entire project.</td>
</tr>
<tr>
<td>Time underestimation</td>
<td>Time underestimation can be a result of many other factors and/or poor</td>
<td>This factor is easy to detect and if it will appear at the initial stages -</td>
</tr>
<tr>
<td></td>
<td>planning, poor developers team management etc.</td>
<td>it is a point to stop the project or to make a decision concerning the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>team management.</td>
</tr>
<tr>
<td>Overbudgeting</td>
<td>Overbudgeting is the most common factor in any size of project. Nearly 90%</td>
<td>Overbudgetting is one of the top reasons for project failure. PM should</td>
</tr>
<tr>
<td></td>
<td>of projects overbudgetted. This issue arises from the complexity of IS project planning.</td>
<td>monitor the usage of financial resources and make decisions as soon as first issues are detected.</td>
</tr>
<tr>
<td>Contracting weakness</td>
<td>Factor which should be detected before the project starts. Poor contract</td>
<td>Contract affects the whole project process. In this case contract must be</td>
</tr>
<tr>
<td></td>
<td>definitions, inadequate requirements and other issues are the effects of low</td>
<td>reviewed from a number of points. Contract milestones can be used as the</td>
</tr>
<tr>
<td></td>
<td>quality BA/contract team.</td>
<td>detectors of issues.</td>
</tr>
<tr>
<td>Low user involvement</td>
<td>The user involvement guarantees less issue in the final stages. However the</td>
<td>Specifically for the large projects user involvement should prevent possible failure related to difference in product vision of developers and users.</td>
</tr>
<tr>
<td></td>
<td>involvement process should start with the development because it creates the base for higher level of user satisfaction in the end.</td>
<td></td>
</tr>
<tr>
<td>Poor consulting and support for</td>
<td>Consulting and support usually is part of the larger and national projects with significant numbers of users. The role</td>
<td>The final phase of projects can be overbudgetted and go out of its timeframe</td>
</tr>
</tbody>
</table>
| **The failures of IT projects. Factors and early prediction** | **of consulting and help is to make the final project phase suitable for the users who are ready for the product.** | **with untrained and unsupported group of users.**

| **Poor or changed requirements** | Poor or inadequate requirements should be detected at the initial stage of implementation or even during analysis phase. Changed requirements are usually the result of poor initial requirements. If the changes are significant that it is a clear detection of a very possible failure. | The projects with poor and often changed requirements will guarantee unsuccessful results. At least the project will be underestimated or/and overbudgetted.

| **Incorrectly selected or used SDLC** | SDLC must be carefully chosen by the analytic group based on such factors as the project dimensions, team experiences, PM experience in the same SDLC etc. | Incorrectly selected SDLC is a very possible factor for failure at early stages. PM cannot keep project within the contracted time frame and budget.

| **Incorrectly selected programming languages, database system etc.** | The technology for development must be carefully selected before the development. The main points to consider are the dimensions of the project, the cost of development and the team experience. Early detection of these issues must be based on the review of the requirements and team experience and abilities. | Incorrectly selected technologies pose a significant factor for failure. Developers will definitely not have time to redesign the product.

| **Skipping or oversight of testing phase** | Currently those factors are less common than 10 years ago. However some projects still experience poor or inadequate testing and/or testing methodologies which mismatch the development process, languages etc. | Affects nearly the entire project and finally will become a reason for failure with time underestimation or overbudgetting.

| **Poor risk management of project manager** | Low quality of risk management will not protect the whole project against possible risks. PM should have experience in risk management and keep the risk plan implementation under control. If risk plan was inadequately implemented or not implemented at all, it is a reason for failure, which might be difficult to predict. | Can affect any stage of project implementation with any possible issues. Difficult to be detected until project runs well.

| **Experience of Project manager in similar projects** | PM’s experience in the similar kind of projects is an important factor which secures the project against nearly all types of failure. | Inexperienced PM may not detect looming issues on time.

<p>| <strong>PM’s personal quality</strong> | Effectiveness in management as well as | Ineffective management |</p>
<table>
<thead>
<tr>
<th>and effectiveness</th>
<th>experience, outlook and other personal qualities and skills of a PM support him/her in the detection of possible failures.</th>
<th>has obvious negative influence on the entire process.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team members work experience</td>
<td>The team members should have an experience in the technologies, methodologies and languages which are used in the project. Team usually does not have time to study new developing techniques. This issue is easily detected at the earliest stages.</td>
<td>Using unknown technique, methods etc. by developing team may be a reason of time underestimation, large number of bugs, or even inability to create a product.</td>
</tr>
<tr>
<td>Ineffective organizational structure</td>
<td>The developer company organizational structure should be set up as an effective body. PM role in it is to organize this structure and/or reorganize it during the development. If it is outside PM’s power, the ineffective structure should be considered the reason for failure.</td>
<td>An ineffective structure cannot develop any product. So the effect in this situation is significant.</td>
</tr>
<tr>
<td>Market environment</td>
<td>All those factors are difficult or impossible to manage. PM team should recognize those factors before the project starts and decide: if their influence is or will become significant the project must be cancelled.</td>
<td>The effect is significant for the entire project.</td>
</tr>
<tr>
<td>Economic environment</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Social environment</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Political environment</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Apparently there are many more factors that influence any IS project, more or less. In this report we highlighted the factors that are possible to predict before project starts or in its early phases of implementation. The situation on IS market according to Dr. Paul Dorsey was a disaster in 2005 and is continuing to be rather complex in 2014 - which may be improved with the new studies in the near future. And the important historical mission for the current market players is to accumulate objective information about any finalized project for the researchers and analysts.
Bibliography


