This doc is to prepare for presentation, Q&A.

# Project management

## Why IT project fail?

**Why are IT status reports often so rosy, even for projects in trouble?** Because status isn't reported in objective, observable terms. It's often put in subjective terms like "percent done," or "red," "yellow" or "green." As long as you allow reporting to be done in subjective terms, you're going to get results that could be colored. Instead, you should look at "Is it done or not done?" where done has a clear definition. If you want to use colors, have a clear definition of what those colors mean.   
Another reason is that if your reward system values compliance -- not rocking the boat, looking like you're on schedule -- and it doesn't value the early detection of problems, it leads to rosy status reports.   
The third reason is fear, whether it's fear of looking bad or losing your job. You're not likely to remove fear in the workplace, but if you change the other two factors -- status reports that are objective and observable, plus a rewards system that values the early detection of problems -- then you'll have status reports that are reliable.   
  
**Are some IT projects just plain doomed from the start?** Yes, I still see that. Either the data required for the project doesn't exist anywhere in the company, or the project is out of alignment with the business strategy, or the objective is simply unattainable.   
  
**Why do managers continue to fund losing efforts?** One, they don't have any empirical evidence that the project can't be salvaged. Two, they want to salvage what's already been invested. The third reason is fear - fear of looking bad, fear of losing their job, fear of making a mistake.   
  
**At what point do you decide to kill the project despite the sunk costs?** When you know that it can't possibly succeed. That requires knowing your definition of success. If you know that even if you pour more money into it, it's not going to be worth the benefits that you get out of it, then it's already failed, whether you keep going at it or not.

## Surveys

### 1 Scope of the Survey

A total of 203 telephone interviews were conducted with IT and project managers from the finance, utilities, manufacturing, business services, telecoms and IT services sectors in UK. All the managers interviewed had previously taken the lead in integrating large systems within organizations in the Times Top 100.

**Project Evaluation Criteria**

The main IT project failure criteria identified by the IT and project managers were:

|  |  |
| --- | --- |
|  | missed deadlines (75%) |
| bullet | exceeded budget (55%) |
| bullet | poor communications (40%) |
| bullet | inability to meet project requirements (37%). |

The main success criteria identified were :

|  |  |
| --- | --- |
| bullet | meeting milestones (51%) |
| bullet | maintaining the required quality levels (32%) |
| bullet | meeting the budget (31%) |

### 2 KPMG

This study, conducted by KPMG Canada, has been mentioned in the statistics over [IT project failure rate](http://www.it-cortex.com/Stat_Failure_Rate.htm).

**Key Findings**

The main causes of project failure that were identified were:

1. **Poor project planning**. Specifically, inadequate risk management and a weak project plan. Risk management becomes more important as the organization gets bigger, so larger organizations need to pay more attention to this area.
2. **Weak business case**. The need for the system should be justified in ways that relate directly to the organization's business needs.
3. **Lack of top management involvement and support**. This often dooms the project to failure before it starts. Securing buy-in from the top, often by a strong business case backed up with a realistic project plan, is an essential step.

**Additional findings**

|  |  |
| --- | --- |
| bullet | Projects fail more often because of schedule overruns than budget overruns. |
| bullet | Many projects fail because they use new or unproven technology. |
| bullet | Poor estimates or weak definitions of requirements at the project planning stage also contribute to project failure. |
| bullet | Projects can run into trouble due to the vendors' inability to meet commitments. |
| bullet | 60 % of the failed projects were planned to take less than one year to complete. |

### 3 Project Evaluation Criteria

The main IT project failure criteria considered were:

|  |  |
| --- | --- |
| bullet | Cost Overruns |
| bullet | Time Overruns |
| bullet | Content Deficiencies |

**Key Findings**

Opinions about why projects are impaired and ultimately cancelled rank **incomplete requirements** and **lack of user involvement** at the top of the list.

|  |  |  |
| --- | --- | --- |
| **Project Impaired Factors** | | **% of  the Responses** |
| **1.** | **Incomplete Requirements** | **13.1%** |
| **2.** | **Lack of User Involvement** | **12.4%** |
| **3.** | **Lack of Resources** | **10.6%** |
| **4.** | **Unrealistic Expectations** | **9.9%** |
| **5.** | **Lack of Executive Support** | **9.3%** |
| **6.** | **Changing Requirements & Specifications** | **8.7%** |
| **7.** | **Lack of Planning** | **8.1%** |
| **8.** | **Didn't Need It Any Longer** | **7.5%** |
| **9.** | **Lack of IT Management** | **6.2%** |
| **10.** | **Technology Illiteracy** | **4.3%** |
| **11.** | **Other** | **9.9%** |

## We Have a Beta Program ??

Some people feel that the best way to debug a system is to ship it to your customers and wait for trouble tickets. I don't know how things work in your industry, but as far as I'm concerned, finding new customers is hard enough. I definitely don't want to make matters worse by shipping them buggy software.

I've spent the last 5 years in the telecommunications industry where the standard for reliability is 5-9. If I even suggested to my customers that my software was in its Beta phase, they'd hang up on me immediately. I'm positive the same is true in any industry.

Imagine if I published articles that were not only full of grammatical errors but also missed a complete sentence here and there. Would you bother sending me an email to inform me that my articles are defective? Would you recommend my newsletter to your peers? I didn't think so! So imagine how your customers feel when mission critical software breaks and crashes on them.

**Cover all the bases with the knowledge areas:**

1. Scope. After defining scope clearly, teach the cost of changes to reduce change requests, then manage all changes, adding to the project only when it is essential.
2. Time and cost. Use unbiased, accurate estimation techniques. Set up systems to gather, track, and analyze time and cost information, so you can keep them under control
3. Quality. Focus on quality at all three levels to ensure value. At the technical level, trace requirements and design checking and testing throughout the project to reduce errors. Then design a test bed, and implement the tests. At the project level, work to prevent error, then find and eliminate the errors that slipped through. Do as much testing as you can as early as you can. Allow time for rework and retesting to ensure you’ve eliminated errors without letting new ones creep in. At the business level, include customers in testing, and remember that the goals are customer delight and added value.
4. Risk. Plan for uncertainty; prepare for the unexpected. Perform risk management with your team every week of the project.
5. Human Resources. Help each team member step up in self-management and technical expertise. Teach everyone PDCA so that they can improve. Then teach them to work together, until you have a great team of great people.
6. Procurement. Get the supplies and resources you need. If your project involves contracts, be sure to keep the contracts in alignment with project value and specifications, not just generally associated with goals and work.
7. Communications. Have a communications plan, and follow it so that you are in touch with all stakeholders throughout the project. Make sure everyone knows what they need to know to make decisions and get work done. Analyze status information to create status reports. Be prompt and decisive.
8. Integration. Constantly direct corrective action. Evaluate all events that could change the project schedule, and all scope change requests. Review the effects of any change on all nine areas before making a decision, and then implement a revised plan with rebaselining.
9. **Requirements:** Unclear, lack of agreement, lack of priority, contradictory, ambiguous, imprecise.
10. **Resources:** Lack of resources, resource conflicts, turnover of key resources, poor planning.
11. **Schedules:** Too tight, unrealistic, overly optimistic.
12. **Planning:** Based on insufficient data, missing items, insufficient details, poor estimates.
13. **Risks:** Unidentified or assumed, not managed.

[CIO.com cites a Dynamic Markets survey](http://advice.cio.com/remi/two_reasons_why_it_projects_continue_to_fail) of 800 IT managers, reporting that 62 percent of IT projects fail to meet their schedules. Other data:

* 49 percent suffered budget overruns
* 47 percent had higher-than-expected maintenance costs, and
* 41 percent failed to deliver the expected business value and ROI

# Portal/Portlet

Every business needs a website today, but a static web site is not much better than posting a sign with hopes that someone will see it and follow-up in finding you. With today's technology, even the smallest business can easily obtain a dynamic, business class web site or portal, improving sales and customer satisfaction.

With current technologies, our Web Portal / Content Management System can be implemented in less than half the time it takes to develop a custom web site, and the system can most likely be implemented at a lower cost!

A portal is a Web site or service that offers broad range of resources and services like e-mail, forums, search engines, on-line shopping, news, weather information, stock quotes, etc. Portal is a term generally synonymous with the terms gateway or grand entrance into the Internet for many users.

A Web portal software allows aggregation of several back-end systems, processes, sites etc brought together through a single portal page. Portals also provide additional services such as single sign-on security, customization (i.e. personalization) etc.

I think of a portal as the aggregater of a number of disparate applications in a highly personalised manner.

Many of the technologies listed above seem to be available to me already without using a portal.

For instance:

-Single Sign On I could do with a valve in Tomcat

-JAAS support is included with most EE app servers

-Themes/Layouts I could do with templating

-User management "could" be done through the tools included with my security store (User Management for Active Directory, LDAP tools for an LDAP repo, etc)

-Content Management I could get with Jackrabbit or some other higher end framework

Basically, it seems that I'm hearing that the value of portal tech is that it combines a lot of these technologies into one place that you might have to cobble together yourself, or write glue for. Also, the ability to personalize the functions that a user has access to seems to be important.

I think the main advantage of a portal is the standard itself. With a portlet, I get all of that, plus it will run in any JSR 168 compliant portal.

For example, I believe the iGoogle page uses the liferay portal, and have tons of third party portlets to choose from.

Portlets are Web-based components managed by portlet containers that supply dynamic content. Portals employ portlets as pluggable user-interface components—a presentation layer—for information systems. The Java Portlet Specification achieves interoperability among portlets and portals by defining the APIs for portlets and by standardizing the rules for preferences, user data, portlet requests and responses, deployment, packaging, and security. Java Portlet Specification 1.0, Java Specification Request (JSR) 168, was released in October 2003.

In February 2006, the JSR 286 Expert Group was formed to start work on Java Portlet Specification 2.0. When that is finalized, backward compatibility will be in place: JSR 168 portlets will be able to run seamlessly in JSR 286 portlet containers. No recompilation will be necessary.

# Remaining Issues

## Review Cost

Có thưởng, add vào đâu ? budget or deduct ?

Có bị trừ lương ?

Trường hợp nào giảm chi phí nữa ?

## 2. End of project, --> end date of assignment ?