Software project failure – what’s killing us and what we can do about it

Edwin Dando
Assurity, New Zealand

Advancing Expertise in Software Testing
Why study software project failure?

- We all know software projects are often challenged right?
- But just how challenged are they?
- And are NZ projects any more challenged than others?
- And if so why?

Curiosity got the better of me....

Aim of the study

- Provide up-to-date info about failure patterns in software development projects in NZ
- Ali Baigy – my Persian Otago MBA friend
- 16 weeks
- 28 organizations including
  - > 18 software companies
  - > 3 universities
Why bother?

1. Knowledge of software project failure in NZ is limited, out-of-date, incomplete, or irrelevant

2. Our industry is in a crisis
   - People believe they can demand it therefore it can be done
   - Under pressure, developers cut quality to support this belief
   - The results include
     - Developers and customers resenting the profession
     - Failing products, failing companies, and hateful work

*We are there now*
Approach

• Review international studies
  > What are the common project failure modes from overseas?
  > Guide questions to see if we are the same

• Approach people
  > Call first to consider a recent project in their org
  > Either send questionnaire or do over phone

The Participants
Number of FTEs

Revenue
Success rates

- Failure: 60%
- In Between: 20%
- Success: 20%
The majority of the Custom Software Development (CSD) projects failed

Failure in their words...

• “A failure because although the delivery was on time, and to budget, the business did not take it up and failed as a consequence.”

• “It earned our company a lot of money, but disrupted our staffing, was unpleasant to work on, was not fully used by the client and was largely irrelevant to the company as a whole.”

• “From a company perspective it was a success; it has generated sales and paved the way for us to enter new markets. From a development standpoint it was a failure; it was significantly late and of lower quality than we expect.”
What is success?

- From an IT provider: "Yes, the project was a success because we carried out the clients specs."
- but when we dug deeper: "The project sponsor didn't like the end result."

But this is often normal, right?
We carried out the clients specs

But hold on, we already know that
• 35% of requirements change throughout project
• 33-66% of requirements are seldom or never used
The client doesn't know what they want.
Isn't just following their specs a bit irresponsible?

So what are people losing sleep over?
Major concerns

Senior managers most worried about cost

- Support Manager
- Quality Manager
- Project manager
- Managing Director
- IS Manager
- GM Operations
- Development Manager
- Developer Team Leader
- Consultant

Support and QA managers most worried about quality & scope

For Consultant time & cost most important

For Dev Lead time much more important than quality

No surprises here?

Challenges

On failed projects, People, Communication and Scope = biggest challenges
On successful projects, Time, Cost, Estimation & Scope = biggest challenges
Business case?

- Failed projects 27%
- Successful 57%
- In between 67%

Could the team specify software purpose & use?

- 100% on successful projects
- 50% on failed projects
Unmanaged scope change contributes significantly to failure

Failed projects cost twice as much & take 110% longer than planned
### Planning

- Importance of planning in all categories was moderate to high
- A lack of planning was not a failure factor for the projects in this study

### Importance of Leadership

- “There was uncertainty over who was responsible for what, largely due to our manager not being able to give the project as much attention as required.”
- “Leadership was part of the problem. A sponsor with unrealistic expectations and a PM managing a number of projects and really just administrating.”
- “Inexperienced project management and lack of a true project sponsor did not allow the outputs of the project to be worked through with the end users.”
- “The leadership is strong - but allows the individuals to contribute and challenge the leadership and how the project is run. Discussion is encouraged.”
Importance of management support

Increased management support tends to result in higher chance of success.

Development team expertise

More experienced team tends to equal higher chance of success.
Learning

- < 15% failed projects had regular learning process throughout the project.
- Successful projects tended to have a learning process
- Lack of a learning process can be considered a contributing factor.

11 Key Findings

1. Software projects fail in all types of organisations, regardless of size, income, location, number of projects per annum
2. 54% of projects in the study were failures and failed projects cost twice as much & take 110% longer than planned
3. Custom software development is vastly (10 times) more prone to failure than integrating existing software
4. Concern areas
   > Senior managers – cost
   > Quality and Support managers – scope and quality
   > Development Managers – time
   > Consultants – time and cost
11 Key Findings

5. Biggest challenges
   > Failed projects - People, Communication and Scope
   > Successful projects - Time, Cost, Estimation and Scope

6. Most failed projects don’t have a business case whereas most successful ones do

7. Scope clarity
   > On 100% of successful projects, the development team could specify the purpose & usage of the software
   > In less than 50% of failed projects the development team could specify the purpose & usage of the software

8. On failed projects scope changed significantly and not managing this scope change contributed significantly to failure

9. We don’t tend to have good processes to learn from our mistakes

10. More planning doesn’t make any difference

11. Leadership is critical
The 5 Core Characteristics

1. Majority (54%) fail, cost twice as much, take 110% longer and don’t tend to have a business case
2. Biggest challenges are People, Communication and Scope
3. Scope
   - Dev team couldn’t specify purpose & usage of the software
   - Scope changes significantly
   - Poor processes for managing scope change
   - Planning more doesn’t help
4. Custom dev = is 10 x more prone to failure than integrating
5. Lack leadership and management buy in
Lets go deeper on that...

Sanity Check...

• Cross reference against fresh NZ study
  > *Software Development Practices in New Zealand*, The University of Auckland Computer Science Department, Feb 2012 - Diana Kirk and Ewan Tempero

• And their findings?
She’ll be right Trev

• “Issues with practices relating to quality”
  > ineffective or missing design-and-code-checking practices (e.g. reviews, unit tests)
  > a lack of independent testing
  > insufficient separation of environments
  > major issues with clarity and accessibility of requirements.

“Represents a possible gap endemic in the NZ approach to developing software”

Process

• “Organisations do not follow standard models such as Waterfall, Scrum or XP, but rather adapt practices”

• “A majority of respondents claimed to be ‘agile’, but close informal contact with customers was not practiced”

“Individuals do not follow practices in a consistent way. The risk of gaps is clearly large in the New Zealand context.”
So lets make that 7 Core Characteristics...

1. Majority (54%) fail, cost twice as much, take 110% longer and don’t tend to have a business case
2. Biggest challenges are People, Communication and Scope
3. Scope
   > Dev team couldn’t specify purpose & usage of the software
   > Scope changes significantly
   > Poor processes for managing scope change
   > Planning more doesn’t make any difference
4. Custom dev = is 10 x more prone to failure than integrating
5. Lack leadership and management buy in
6. Issues with practices relating to quality
7. Organisations do not follow standard models but rather adapt practices

Back to the investigation...
#1 - Majority (54%) fail, cost twice as much, take 110% longer & don’t have a business case

- Consistent with Product Ownership “Achilles heel”
  - Why are we doing the project?
  - What value does each feature add?
  - Should we continue this project?
  - How can we trim the tail to get an early working version into customer hands?

- Consistent with my own observations

- One solution – quality Product Ownership
  - Work with customers + people developing software
  - Work based on value
  - Validate assumptions and reduce risk via regular feedback cycles

# 2 People, Communications & Scope

- And more up front planning doesn’t help....

Planning

- Success
- In Between
- Failure

0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5
# 2 People, Communication and Scope

- But done properly agile nails people, communication and scope!
- Oh, that’s right: “organisations do not follow standard models but rather adapt practices”
- Epic Fail

# 3 - Scope

- Agile gives you
  > Ability to change scope frequently
  > Stronger people focus through self-management and empowerment
  > Increased communications

All of this for free!!!
The Sprint Agreement

<table>
<thead>
<tr>
<th>The Development Team</th>
<th>Clients</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Every Sprint you can have us do something new as you see fit.”</td>
<td>“We will leave you alone let you work on what we need most.”</td>
</tr>
</tbody>
</table>

FLEXIBILITY

STABILITY

# 4 custom dev = is 10 x more prone to failure than integrating

- But this is where the magic happens!
- Innovative products aren’t the result of integration
- Did Steve Jobs integrate existing products?
- Did Yikebike integrate existing products?
- Did Lars & Jens Rasmussen integrate – no they built!
- We have a system that fosters innovation while managing complexity and controlling risk....
# 5 Leadership & Management Buy In

- Servant Leadership – behaviours are values in action
- Predictive management versus empirical management
- The desire to appear certain vs. the courage to admit software is not predictable and risk is best managed incrementally, with data
- Authority moves down the organization – trust your people!

NZ/AU - Strengths to Opportunities

- Use our isolation, innovation, attitude and independence to lead.
- Harness our pioneering innovation

The changing role of management

- People don’t set out to fail each day - they set out to succeed
- Successful managers
  > create an environment for success
  > empower their people
  > stand aside
- Define the "what" not the "how"
- Accept that people don’t work for money, they work for AMP
  > Autonomy, Mastery, Purpose
- The world is changing and Gen-Y are helping challenge the status quo
- Manage them as you would a team of volunteers
# 6 - Issues with practices relating to quality

- missing design-and-code-checking practices
- a lack of independent testing
- insufficient separation of environments
- major issues with clarity and accessibility of requirements.

- Agile will highlight these insufficiencies quickly and often (like your mother in law coming to stay – you will be reminded of your deficiencies every day…)

“Scrum will highlight every deficiency and impediment that the enterprise has so the enterprise can fix them and change into the best organization in its market”

Ken Schwaber, Scrum is Hard & Disruptive, 2006

# 7 - Organisations don’t follow standard models - they adapt practices

- The practices exist for good reason. Don’t skip them until you have mastered the basics

守
The student follows the teachings precisely. Concentrates on the task, without worrying about theory.

破
Basic practices working - student starts to understand the underlying principles & branch out.

離
Student isn’t learning from other people, rather own practice. Creates own approaches & adapts learning to circumstances

“Scrum is not a methodology that needs enhancing. That is how we got into trouble in the first place, thinking that the problem was not having a perfect methodology. Effort centres on the changes in the enterprise that are needed”

- Ken Schwaber, Scrum is Hard & Disruptive, 2006
Finally - Complexity

- The basic problem lies in our comprehension of what complexity is.
- Paradoxically, the deeper we look into complexity the more we see.
- As a result, we now appreciate that prediction of complex systems is (probably) impossible.

*Newcastle University*
*Engineering Design Centre*

Contrasting Project Failure

- **New Zealand**
  - We succeed less
  - We fail more
  - We have less in between

Summary

We are failing for silly reasons and we know better

- **Product Ownership** - business case, communication, clear view of product purpose & usage, ability to change and manage scope
- **People** deliver projects – team need a clear view of the product purpose & usage, plus the ability to get involved in this together
- **Leadership is critical** – you are only a leader if people follow
- **Values trump everything** – an open “safe to fail” culture wins
- **Journey of constant improvement** - effort centres on the changes in the enterprise
- **Start** with the practices and only adapt once you know what you are doing
- **Don’t be scared of custom dev** – just actively manage the risk using transparency and empirical processes

The future...

- Failure isn't bad, as long as we learn from it
- Our industry is a wonderful industry full of amazing people who want to succeed
- But we must be brave, creative and honest – past management thinking is failing us
- We can use isolation to reduce negative influences to follow and lead in our own way
Questions?