Agile software development and benefits management: A perfect match?

Magne Jørgensen
Simula Metropolitan, Scienta, Kathmandu University

Based on:


• Huge investments in digitalization. What does it give us in return? Keynote Software 2018 (DnD’s annual conference, Oslo, Norway).
Q: Why do we invest in software development?

A: To create client benefits, value for the stakeholders, positive effects for the users, achieve goals, ....

Q: What has been the focus of software project management for most of its history?

A: Delivering the specified functionality on cost and within the budget. Managing risks, such as scope creep and unstable requirement. Not much about benefits management here ...
Q: What has usually happened when a project is described as a failure?

A: It has had a large cost or time overrun (or being cancelled)

To illustrate how strangely we tended (and still tend) to think about failures, see the following statistics (from PMI, 2019)
Where is benefits management in this statistics? (change management?)

Is much requirement change and change in priorities a threat or an opportunity for software development work?

More on this later ...
(but a hint is that «it depends»)
Software development leads to greater benefits and is more important for our welfare and happiness than most are aware of.

… 10% increase in ICT investment leads to, on average, to a 0.6% (percentage points) increase in productivity (historically a productivity increase higher than when introducing steam engines and at the level of introducing electricity).

"… the productivity increase impact of ICT has grown over time."

Around half of the current increase in productivity is due to ICT-investments!
Wealth is strongly correlated with happiness «World happiness report»

Figure 2.2: Ranking of Happiness 2014-2016 (Part 1)

1. Norway (7.57)
2. Denmark (7.52)
3. Iceland (7.50)
4. Switzerland (7.44)
5. Finland (7.46)
6. Netherlands (7.37)
7. Canada (7.36)
8. New Zealand (7.34)
9. Australia (7.28)
10. Sweden (7.28)

- Explained by: GDP per capita
- Explained by: generosity
- Explained by: social support
- Explained by: perceptions of corruption
- Explained by: healthy life expectancy
- Explained by: freedom to make life choices
- Explained by: longevity

Self-reported happiness vs income over time

The vertical axis shows the share of people who say they are ‘very happy’ or ‘rather happy’. The horizontal axis measures average national income. Each country is drawn as a line joining first and last available observations.

- Income is key for happiness.
- Productivity increase is the key income increasing factor.
- Technology is the key productivity driver.
- IT is (today) the key technology driving productivity.
What is benefits management in software development?

A set of processes:
– Identify, prioritize and estimate benefits (and costs)
– Develop a plan for when and how to realize benefits
– Allocate responsible people for the realization of the benefits
– Continuous delivery, feedback, learning, re-prioritization and management (even realization) of benefits during the project execution
– Evaluation of realized benefits

What is agile in software development?

«Agile is the ability to create and respond to change. It is a way of dealing with, and ultimately succeeding in, an uncertain and turbulent environment». (Agile Alliance)
12 agile principles

• Customer satisfaction through early and continuous software delivery
• Accommodate changing requirements throughout the development process
• Frequent delivery of working software
• Collaboration between the business stakeholders and developers throughout the project
• Support, trust, and motivate the people involved
• Enable face-to-face interactions
• Working software is the primary measure of progress
• Agile processes to support a consistent development pace
• Attention to technical detail and design enhances agility
• Simplicity
• Self-organizing teams encourage great architectures, requirements, and designs
• Regular reflections on how to become more effective

Claim: There are better opportunities for good benefits management when software development is agile
Our studies on benefits management and agile software developments:

• Nine surveys, with 50-200 participants each, representing around 1000 Norwegian software projects in the public and the private sector.
• In-depth, interview-based examination (case studies) of 35 software projects in the public sector of Norway
• Ongoing studies in two large organization on benefits management in large scale agile

Success and failure rates found in our studies

All studies give similar results:
• Around 50-60% projects were acceptable, successful or very successful on benefits, cost control, time control, productivity and quality
• Around 30-40% were problematic (but not failed) on least one factor.
• Around 10% projects failed (either cancelled or completed with little benefits)
It helped to work agile (especially related to delivering client benefits), but …

... only when agile included frequent delivery to production and flexible scope.

Agile projects not including these two practices were LESS successful than non-agile projects in terms of client benefits! These two practices are strongly connected to benefits management.

Similar results in our follow-up surveys and studies

<table>
<thead>
<tr>
<th>Agile</th>
<th>Frequent delivery to production</th>
<th>Flexible scope</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client benefits</td>
<td>16%</td>
<td>22%</td>
</tr>
<tr>
<td>Technical quality</td>
<td>21%</td>
<td>6%</td>
</tr>
<tr>
<td>Budget control</td>
<td>2%</td>
<td>22%</td>
</tr>
<tr>
<td>Time control</td>
<td>8%</td>
<td>11%</td>
</tr>
<tr>
<td>Efficiency</td>
<td>11%</td>
<td>5%</td>
</tr>
</tbody>
</table>

Benefits management helps, especially during the project execution …

Survey 1:

<table>
<thead>
<tr>
<th>Benefit management practices</th>
<th>Proportion</th>
<th>Increase in success rate (wrt benefits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost-benefit analysis (up front)</td>
<td>47%</td>
<td>6%</td>
</tr>
<tr>
<td>Benefit responsible appointed</td>
<td>57%</td>
<td>22%</td>
</tr>
<tr>
<td>Plan for benefit management</td>
<td>33%</td>
<td>31%</td>
</tr>
<tr>
<td>Benefit management during proj. execution</td>
<td>53%</td>
<td>34%</td>
</tr>
<tr>
<td>Evaluation of benefit during/after proj. exec.</td>
<td>31%</td>
<td>19%</td>
</tr>
</tbody>
</table>

Survey 2 (in-depth study):

<table>
<thead>
<tr>
<th>Benefit management practices</th>
<th>Present</th>
<th>Not present/don’t know</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost-benefit analysis (up front)</td>
<td>31% with problems</td>
<td>22% with problems</td>
</tr>
<tr>
<td>Benefit responsible appointed</td>
<td>28% with problems</td>
<td>29% with problems</td>
</tr>
<tr>
<td>Plan for benefit management</td>
<td>29% with problems</td>
<td>28% with problems</td>
</tr>
<tr>
<td>Benefit management during proj. execution</td>
<td>20% with problems</td>
<td>30% with problems</td>
</tr>
</tbody>
</table>
Successful benefits management in a changing world required agile development with frequent deliveries (very large difference in success rate!)

Agile software projects with benefits management practices during project execution were less affected by large project size.
Time & material type of contracts were much better in terms of achieved client benefits. That is (at least partly) due to benefits its connection between agile and benefits management.

<table>
<thead>
<tr>
<th></th>
<th>Fixed price</th>
<th>Time &amp; Material</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client benefits</td>
<td>0% (success rate)</td>
<td>59%</td>
</tr>
<tr>
<td>Technical quality</td>
<td>22%</td>
<td>24%</td>
</tr>
<tr>
<td>Budget control</td>
<td>33%</td>
<td>31%</td>
</tr>
<tr>
<td>Time control</td>
<td>11%</td>
<td>29%</td>
</tr>
<tr>
<td>Efficiency</td>
<td>0%</td>
<td>19%</td>
</tr>
</tbody>
</table>

Success pattern: Interaction between contract, agile and benefits management …

- Stronger emphasis on evaluation of skill, less emphasis on low price, in selection of provider
- Stronger client involvement in management (monitoring, selection) of resources
- Project scope changes and scope flexibility perceived as an opportunity
- More use of agile development with frequent deliveries to production and flexible scope
- More focus on benefit management during the project execution
- More, earlier and better feedback from users and other stakeholder
- Higher likelihood of project success
- Higher likelihood of delivering the expected client benefits
- Higher likelihood of competent provider and skilled developers
- Higher likelihood of good quality and productivity
- Less risk of opportunistic behaviour of provider
- Stronger client and stakeholder involvement in project management
- Higher likelihood of competent provider and skilled developers
- Higher likelihood of delivering the expected client benefits
- Higher likelihood of good quality and productivity
- Less risk of opportunistic behaviour of provider
- Stronger client and stakeholder involvement in project management
- Higher likelihood of project success
Project scope changes and scope flexibility perceived as an opportunity

Stronger client and stakeholder involvement in project management

More use of agile development with frequent deliveries to production and flexible scope

Higher likelihood of good quality and productivity

Higher likelihood of competent provider and skilled developers

Higher likelihood of delivering the expected client benefits

Higher likelihood of project success
Success pattern: Interaction between contract, agile and benefits management …

- Stronger emphasis on evaluation of skill, less emphasis on low price, in selection of provider
- Stronger client involvement in management (monitoring, selection) of resources
- Project scope changes and scope flexibility perceived as an opportunity
- Stronger client and stakeholder involvement in project management
- More use of agile development with frequent deliveries to production and flexible scope
- More, earlier and better feedback from users and other stakeholders
- Higher likelihood of project success

Failure pattern: Interaction between contract, agile and benefits management

- Stronger emphasis on low price in selection of provider
- Lower emphasis on provider skill
- Lower client involvement in management of resources
- Project scope changes and scope flexibility perceived more as a risk
- Lower client/stakeholder involvement in project management
- Less use of agile development with frequent deliveries to production and flexible scope
- Less focus on benefit management during the project execution
- Less and late feedback from users and stakeholder
- Higher risk of opportunistic provider behaviour, when making financial loss
- Higher risk of opportunistic provider and developer skill and productivity problems
- Higher risk of provider and developer skill problems
- Higher risk of client benefits problems
- Higher risk of project failures
- Stronger emphasis on low price in selection of provider
- Lower emphasis on provider skill
- Lower client involvement in management of resources

- Project scope changes and scope flexibility perceived more as a risk
- Stronger focus on specification and less on what gives the client more benefits
- Lower client/stakeholder involvement in project management
- Less use of agile development with frequent deliveries to production and flexible scope
Conclusions

- There are success and failure patterns, not isolated success and failure factors
- Agile development, with its frequent deliveries and flexibility in scope, enables good benefits management during project execution
- Other factors, especially choice of contract, supports or limits the ability to implement good benefits management practices in agile development.
- Agile software development is essential to enable successful benefits realization.
What are key characteristics of software development?

- Exposed to a **fast-changing world** (technology, needs, opportunities)
- Producing **innovations** (never constructing the same twice)
- **Transformation** projects (change of work processes)
- Enables **agility** (such as scope flexibility, less upfront planning and specification work, frequent deliveries, late changes)
- **Continuous development** (the organization of software development work as a project is by more and more software professionals believed to problematic.)