Agile Contracting
The challenges of velocity and quality
October 2019
Eric van der Vliet
Agile Contracting – The Challenges

Agenda

• Why Agile?
• What is Agile?
• Velocity and Quality
• Velocity and Quality in Agile contracts
• Conclusions
Why Agile?
Why Agile?
Traditional lifecycles

Plan  Execute

Business value

Planned
Actual

Stop

Business value

© 2019 CGI Nederland B.V.
Why Agile?

Agile deliveries are value driven

- **Waterfall**
  - Fixed
  - Estimate
  - Scope
  - Plan Driven
  - Costs
  - Deadline

- **Agile**
  - Costs
  - Value / Vision Driven
  - Scope
  - Deadline
What is Agile?
What is Agile?

A philosophy

• A way of organizing yourself to achieve flexibility with respect to business objectives
• Agility means anticipation on change and focusing on value
• Flexibility is required due to a constantly, rapidly changing world around us

• Agile is not Scrum or Kanban

• Moving to an Agile way of working requires a fundamental change of:
  • Culture
  • Behavior
  • Attitude

• Agile development relies on traditional software development fundamentals
  • just approaches them to focus on value
The Agile Manifesto
Value on the right; Value on the left

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- Individuals and interactions
- Working Software
- client collaboration
- Responding to change

While there is value in the items on the right, we value the items on the left more.
Velocity and Quality
Except Value and Velocity is monitoring of the Quality important
Budget & Quality in SAFe®
Built-in software quality

“Reproduced with permission from © 2011-2017 Scaled Agile, Inc. All rights reserved.”
Define, Measure, Improve

What is not **defined** cannot be **controlled**

What is not **controlled** cannot be **measured**

What is not **measured** cannot be **improved**

**Define, Measure, Improve**

- Velocity and Quality measurement
- Development budget
- Product Quality
- Achievement of Sprint and PI targets
- (Definition of Done)

- Team Velocity
- Product Quality
Definition of Done
Ready for UAT or Production

Example of Definition of Done:
• Code is peer-reviewed and refactored
• Code is deployed to test environment
• Code has passed the static code analysis
• Feature is tested against acceptance criteria
• Feature passes regression testing
• Feature passes smoke test
• Feature passes performance test
• Minor defects logged within product backlog for prioritization
• Feature is documented / Product backlog updated with notes and documentation
• Feature ok-ed by UX designer
• Feature demonstrated in Sprint review
• Feature/Stories accepted and signed off by Product Owner

A shared understanding of which (acceptance) criteria a feature must satisfy to be releasable. They have to provide value and they have to meet a certain quality standard
Velocity & Quality measurement - 2

Collection of (historical) data

- Collaboration with Product Owner (PO)
- Assumptions
- Clarification
- Acceptance criteria

Business

Define

Source Repository

Measure

Continuous Integration

Improve

Design & implementation

- Code Check in / out
- Unit Testing
- Packaging
- Integration Testing
- Compiling
- Code Quality Verification
- Deployment
- Handover
Velocity and Architecture - 1
Built in quality due to (right) architectural decisions

Plan -> Execute

Business value

STOP

Architecture
Velocity and Architecture - 2

Built in quality due to (right) architectural decisions
Velocity and quality in Agile contracts
Velocity and Quality in Agile contracts

Contracting

“If we can continuously ensure we are delivering on our promises, the client is successful, and we’re going to be successful too.”

- The contract is a **legally binding agreement** which recognizes and governs the rights and the duties of the parties to the agreement.

- The contract describes the **terms and conditions** defined for both the client and supplier to enable a successful delivery.

- The contract describes the **scope of supply** based on the client demand translated by the supplier in a proposed solution.
Contract and Proposal aspects

Contractual agreements
- Controlling specifications
- Governance and dispute resolution
- Budget and Payment
- Reporting / KPIs
- Acceptance
- Changes to supply
- Intellectual property
- Warranties and Indemnities
- Liability
- Termination

Delivery objectives
- Business value
- Client satisfaction
- Financial health

Supplier abilities and skills
- Domain expertise
- Solution
- Resources
- Delivery strategy
- Planning
- Cost estimation

Client demand
- Scope
- Budget
- Schedule
- Quality

Risks
- Scope
- Schedule
- Staffing
- Financial
- Quality
Velocity and Quality in Agile contracts

Collaboration & interaction

Delivery

Scope / Specs

Develop

Validation / Acceptance

Governance

Product Owner

Development Team and Scrum Master

Business owner

Engagement manager

Dashboard
Estimation

Ensure we are delivering our promises

Scope has been defined but

• What will fit in a sprint?
• What will fit in an increment?

Sizing of the scope

• Features are on a high level
• Stories are one liners; As a <role> I want <…..> such that <……>}

Size has been defined but

• Size in Function Points (FP’s) and Story Points (SP’s)
• What will be the team productivity (hr / FP’s)
• What will be the team velocity (SP / Sprint)?
• How representative is historical data?
Team efficiency

Depends on the quality of the process, products as well as the team optimization

- Team efficiency depends on the quality of the sprint backlog
- Product management is responsible for providing the stories based on the defined features
- The product owner will work with the team to clarify the user stories
- User stories must be detailed enough to be able to make an accurate estimate for the sprint
- A Definition of Ready is applied to verify if the user stories are detailed enough
- A Definition of Done is applied on the sprint results
Optimal team performance requires time

Team performance will increase over time

Tuckman's Team & Group Development Model

Forming → Storming → Norming → Performing

Effectiveness of Team vs. Performance of Team
Definition of Ready

Ready for realisation by the team

- Example of Definition of Ready:
  - Description clearly articulates the role, action and benefit
  - Acceptance criteria clearly defined
  - User Experience requirements and artifacts (e.g. wire frames) included
  - Supporting documents (e.g. business rules) referenced and/or included
  - User Permissions defined (if applicable)
  - Performance criteria defined (if applicable)
  - Mapped to a Feature and classified as parity or enhancement
  - Product Owner identified and has approved the user story
  - Development team has reviewed and confirmed they understand
  - Includes initial estimation (in story points) of complexity
  - Can be finished in a single sprint
  - Sprint Review demonstration expectations defined

A set of minimum criteria before it’s ready for inclusion in the work of the next sprint, agreed by the Scrum team
# Agile contract matrix

## Contract types vs Contract aspects

<table>
<thead>
<tr>
<th>Maturity</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>High</td>
<td>High</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type of Contract</th>
<th>Controlling specifications</th>
<th>Governance and dispute resolution</th>
<th>Budget and Payment</th>
<th>Reporting / KPIs</th>
<th>Acceptance</th>
<th>Changes to supply</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flexible contract based on team size, basically secondment (T/M) with minimum reporting</td>
<td>high level</td>
<td>limited</td>
<td>hourly rate</td>
<td>indicative per sprint</td>
<td>DoD/ sprint</td>
<td>implicit backlog</td>
</tr>
<tr>
<td>Contract based on a defined team (cost) rate, secondment, including a.o. mutual governance and management, as an effort obligation</td>
<td>high level</td>
<td>full range</td>
<td>per sprint or PI</td>
<td>extensive</td>
<td>DoD/ sprint</td>
<td>implicit backlog</td>
</tr>
<tr>
<td>Unit / Output based contract, all of the above, including KPI's, as a result obligation</td>
<td>high level</td>
<td>full range</td>
<td>per unit (eg. function point)</td>
<td>extensive</td>
<td>DoD/ sprint</td>
<td>implicit backlog</td>
</tr>
<tr>
<td>Scope based contract, (scope is largely fixed, budget is indicative)</td>
<td>detailed</td>
<td>full range</td>
<td>milestones?</td>
<td>extensive</td>
<td>acceptance criteria(!)</td>
<td>Explicit contract change</td>
</tr>
<tr>
<td>&quot;Outcome&quot; based contract, where agile delivery becomes part of a &quot;business process&quot;, with a result obligation</td>
<td>detailed</td>
<td>full range</td>
<td>per achievement?</td>
<td>extensive</td>
<td>outcome?</td>
<td>Explicit contract change</td>
</tr>
</tbody>
</table>
Delivery risk vs Agility / Agile Maturity
Delivery risk vs Agility / Agile Maturity

[Diagram showing the relationship between Agile Maturity levels and Delivery risk]

- Agile Business Maturity
- Agile Portfolio Maturity
- Agile Program Maturity
- Agile Team Maturity

Agility

Organizational level
Conclusion
Conclusion

• Agile is flexible but requires a continuous focus on velocity and quality from a delivery perspective
• Software quality and velocity are important KPI’s in Agile contracting (measurements on team level)
• Contract management becomes important if the delivery is not according to expectations
• The type of contract is depending on the maturity of both the client and the supplier
Questions?
Thank you!

Mail: eric.van.der.Vliet@cgi.com
Linked-in: www.linkedin.com/in/eric-van-der-vliet-8963652/