Agile Software Development: Looking Past the Catchphrases to understand the Contract

John Beardwood
IT.Can Roundtable
February 18, 2016

FASKEN MARTINEAU

Introduction

- A recent survey of development and IT professionals shows that agile is now the norm.
- HP survey of 601 software developers/IT professionals, re their co's:
  - 77%: "pure agile" or "leaning towards agile"
  - 9%: "pure waterfall" or "leaning towards waterfall"
  - 24%: hybrid approach

    [State of Performance Engineering 2015-16: TechBeacon]
- Compare with 2012 Survey: 35% favoured agile
- However, neither customer knowledge nor applicable development contracts have caught up to this reality

Introduction

1. Must be the Right Project
   - **Advocates**: "liberating"; fast, efficient & creative
   - **Detractors**: undisciplined hacking; less robust, buggier s/w
   - However, Agile is neither "good" nor "bad" on its own - depends on the projects for which it is being used: it will not be appropriate for certain projects
2. Must be the Right Contract
   - Waterfall contract framework does not sit comfortably with Agile projects

FASKEN MARTINEAU

Introduction

In this presentation we will review:

1. Waterfall & Agile benefits & challenges: and projects most/least suited for Agile
2. Key issues in Agile contracts, with a view to responding to the risks of Agile
3. Case Study #1: Choosing the Right Project
   - *De Beers UK Limited vs. Atos Origin IT Services UK Limited*
4. Case Study #2: Choosing the Right Contract

FASKEN MARTINEAU

I. Waterfall vs. Agile

FASKEN MARTINEAU

A. Traditional Waterfall

Good way to explain Agile is to contrast with Waterfall = planning, organization & documentation

1. Gather/define business requirements
2. Develop system design specifications (i.e. “roadmap”)
3. Divide project into distinct modules with detailed deliverables, each assigned to different programming team, to programme/test/debug their respective module, per test plans & reports
4. Formal, well-documented change mgmt process
5. Post-coding stage, modules are assembled, tested, and debugged, often with significant time spent rewriting code to ensure integration.

A. Traditional Waterfall

Benefits:

- *Limited, targeted customer involvement*: at beginning (i.e. re requirements development) & end (i.e. final integration & testing).
- *Clear budgeting*: clear specifications/timelines = accurate developer bids/customer budgets
- *Single development roadmap*: large teams, different geographies can follow over longer time horizons.
- *Divisible modules*: facilitates outsourcing/ subcontracting portions of project.
- *Benchmark for change mgmt*: as document focused, can quickly determine impact on timeline/ cost/operation; also, changes are well documented pursuant to change mgmt so can reduce disputes

FASKEN MARTINEAU

A. Traditional Waterfall

Disadvantages:

- *Less flexible/responsive to change*: Customer not always able to clearly express requirements at outset; often, need for certain different/additional deliverables only apparent once development has begun. Rigid planning/detailed documentation of Waterfall does not respond well.
- *More isolated from collegial/customer feedback*: Segmenting the project into modules discourages cohesion & creativity, reduces valuable customer/user input, & often results in design flaws not discovered until testing = integration & acceptance carry a significant portion of operational & legal risk

FASKEN MARTINEAU

B. What is Agile development?

= **iterative**/incremental, with **less documentation** & formality, and **more teamwork** & experimentation

1. *Ongoing/evolving specifications*: reqm'ts are largely emergent & change via discussions with customer.
2. *More extensive/deeper customer involvement*: customer's personnel are on site regularly, collaborating with developers & modifying reqm'ts during project =
   = more customer control but also more commitment.
3. *Smaller workteams*: 2 - 8 people in collaborative "war room".
4. *Well rounded, experienced programmers able to multi-task*: So no person is bottleneck (but also much less replaceable).

FASKEN MARTINEAU

B. What is Agile development?

5. *Ongoing, periodic testing/QC*: increments of 1 to 3 months to allow for quick testing/repairing. Techniques such as continuous integration improve quality/agility & reduce integration risk at project end.
6. *Success measurement*: successful Agile project "will build something different and better than the original plan foresaw." Concern is of course whether really "better".

- Not monolithic, as different kinds of agile: e.g.:
  - Kanban
  - Extreme Programming (XP)
  - Scrum
  - Crystal
  - Dynamic Systems Development Method (DSDM)

C. When is Agile appropriate?

- Agile is **less** appropriate (and “planned” methodologies such as Waterfall are more appropriate) where:
  - Large-scale development efforts.
  - Distributed development efforts (non-co-located teams).
  - Forced adoption of Agile by the development team.
  - Junior developers.
  - High criticality projects - i.e. mission-critical systems where failure is not an option at any cost (software for surgical procedures).
  - Relatively fixed requirements –i.e. the requirements do not change often.
  - Order-based development culture.
  - Only limited degree of customer participation available.

FASKEN MARTINEAU

C. When is Agile appropriate?

- Agile is **more** appropriate in development contexts where:
  - Smaller-scale development efforts.
  - Co-located development efforts.
  - Adoption by an Agile-experienced development team.
  - Senior developers.
  - Low criticality projects - i.e. systems with a higher tolerance for failure..
  - Frequently changing requirements.
  - “Chaos-based” development culture.
  - High degree of customer participation available

FASKEN MARTINEAU

II. Agile Risk Mitigation: Key Contract Issues

FASKEN MARTINEAU

A. Contract required?

- "The central values of all Agile process are: individuals and interactions over processes and tools; working software over comprehensive documentation; customer collaboration over contract negotiation; and responding to change over following a plan." (www.Agilemanifesto.org)
- BUT notwithstanding that one unifying theme between different Agile schools is premium each places on:
  - trust and understanding between the parties, and
  - a minimum amount of documentation & formality…
- …at most basic level customer still has a budget, & requires results.

FASKEN MARTINEAU

A. Contract required?

- Also, whether or not Agile collaborative approach reduces likelihood of disputes, when such a dispute *does* occur, Agile can make resolution more difficult.
- For example:
  - If little or no documentation has been produced, to what documents can the parties point as evidence of their respective expectations?
  - If certain key project staff leave, is there sufficient documentary evidence available to assist successors?
  - For this non-module based methodology, how are payment milestones addressed?
  - Do different forms of Agile raise additional & different kinds of risks?

FASKEN MARTINEAU

B. Process-Oriented Contractual Framework

- Customer will bring initial business req'mts that it wishes to achieve
- BUT, rather than agreeing on specific characteristics for final software deliverable, parties will create a contractual framework for a process that will allow them to both:
  - respond to changing requirements, and
  - reach the end result desired, without actually specifying exactly what that result will be beyond the basic functional requirements.
- Therefore, and as outlined in more detail below, an Agile development contract will necessarily require a greater emphasis on governance.

FASKEN MARTINEAU

C. Is Agile right for you? Buying into the Agile Process

- Both parties need to be well-educated re Agile philosophy, terminology, & each step of Agile process.
- Customer must:
  - prepare for more intense level of participation than in Waterfall projects
  - continuously dedicate several Customer personnel to project
  - ensure dedicated individual(s) are knowledgeable about all aspects of business potentially impacted by project, so can effectively express user requirements as they emerge through development cycle.

D. Doing the due diligence: The First Date

1. Initial Phase One Agmt: allows parties to test the waters & opt out of process after initial phase
2. Due diligence: understand business objectives, scope & req'd dedicated resources.
3. Allow for a non-liability opportunity to opt out from Agile process after due diligence: so can, for example, adopt a Waterfall method of development.

FASKEN MARTINEAU

E. The Relationship: Iterations

- Iterations are key — arguably the key— to any Agile process
- Therefore parties must therefore focus on implementing a governance structure for the iteration process which is both efficient (to maintain agility) and effective (to minimize risk).
- Waterfall risk is allocated to integration & testing phases, vs. Agile risk is scope management through iteration process

FASKEN MARTINEAU

E. The Relationship: Iterations

- Agile process-oriented contract should set out:
  - governance framework for iteration process:
    - frequency of iterations,
    - how iteration meetings are to be conducted
    - process for agreeing on timelines, priorities, & functionality;
  - for each iteration:
    - mandatory functionality
    - estimated development effort for required items
    - consent from both sides regarding the objectives of the iteration

FASKEN MARTINEAU

E. The Relationship: Iterations

Example of application of Agile methodology:

- Delivery approach based on lean and agile practices using Kanban.
  - **Kanban**: Less structured than Scrum. Model for change via incremental improvements. Improves rather than rebuilds existing working processes over time. Uses a Kanban board.
  - **Scrum**: Process framework. Better where org. needs a fundamental shift to more efficient process.
- Key components:
  - **Sprints:** time-boxed iterations from 3 to 4 weeks
  - **Story Map:** built from pre-built stories as an initial baseline
  - **Program MFs & Integrated Testing:** defined set of Minimum Features
  - **Requirements:** developed via stories, use cases, functional technical specifications, supplemented by models, frameworks, prototypes, dashboards, swimlanes, sprints, iterations, & scrum feedback

FASKEN MARTINEAU

F. Testing

- Acceptance testing is also an important aspect of Agile
- BUT, unlike Waterfall, testing occurs over different time horizon:
  - iterations are relatively short
  - deliverables are smaller
  - testing occurs throughout the project,
  - testing will be more informal & may take an hour/a day.
- Agile contract (or program handbook) should outline:
  - testing criteria that clearly set out when iteration is deemed complete
  - consequences of passing & failing an acceptance test: incl. iteration works well independently, but does not integrate properly
  - process for test results to be documented & signed off by representatives of both parties
  - how dispute resolution can be activated re difference of opinion re an acceptance test.

FASKEN MARTINEAU

G. Who is calling the shots?

- Intense collaboration btw developers & customers
- So, important to set out a governance framework which specifies which individuals may make key decisions.
- By def'n scope of an Agile project will evolve, and, at certain points, changes to scope will be made and activities outside the scope will be carried out.
- Which representatives will have the authority to make change?
  - If all team members effect such change = more rapid and creative process, but may entail greater risk of error or disagreement.
  - Reserving such decision-making power to the most senior member(s) of the team reduces risk, but also reduces flexibility.

G. Who is calling the shots?

- Typical Governance Roles:
  1. Customer:
     - Executive Sponsor
     - PAS Steering Committee
     - Business Owner (BO)
     - Active Business Owner (ABO)
  2. Vendor:
     - Delivery Manager
     - Release Lead
     - Scrum Master
     - Program Management Office (PMO)

FASKEN MARTINEAU

H. Documentation & Change Management

- Agile contract is process-oriented: will focus on procedure rather than specifications, as many specifications & related decisions will arise in Scrum meetings of the development team.
- Informal nature of Agile means particularly important to document what changes require what level of agreement
- E.g. Change ledgers vs Change Orders

FASKEN MARTINEAU

I. Sharing the Risk: Fee Structures

- Ideally, the risk should be shared btw the parties:
  - Waterfall project: project plan may incl. series of payment milestones
  - Agile project: similar motivations but:
    - without preset set of specifications or “design roadmap”, difficult to agree in advance on payment milestones; &
    - many argue that Agile development is incompatible with fixed price projects, because successful fixed price engagements are dependent on fixed scope, which Agile projects by def’n do not have
  - BUT customers need to budget, & therefore need some certainty built into the payment structure.

FASKEN MARTINEAU

I. Sharing the Risk: Fee Structures

Alternative fee structures:

- Estimate and fund by iteration only: as each iteration is intended to produce a deliverable (recall: "delivery of working software over documentation"), this manages costs while allowing for more flexible T&M payments per iteration.
- Cost Plus (by iteration): Developer is paid internal (non-margin) only, but delivery bonuses are paid for working software, so developer will not make margin unless delivers

FASKEN MARTINEAU

I. Sharing the Risk: Fee Structures

- Cost-Sharing (by iteration): Any amount by which the actual cost exceeds/is below budget target (or past a specified acceptable percentage above or below) is shared so share cost savings & cost overruns
- ***Earned Value Management ("EVM")***: common PM technique for measuring planned cost, actual cost, & cost against planned performance/value generated (see Case Study #2)

FASKEN MARTINEAU

J. Changing Horses: Exiting Agile Projects

- Agile projects can very difficult to exit, given that:
  - Escrow arrangements: have limited utility given absence of written specifications to deposit, so nature of deposited materials requires greater attention (e.g. include working notes, & contact info for developers).
  - Non-solicitation covenants: lack of documentation means personnel rather than code are the most significant assets = where termination, customer will want developer to waive non-solicitation covenants.
  - Isolation from overall development process: Take advantage of Agile project interactivity by embedding customer "shadow" developers in development teams, & ensuring customer personnel take comprehensive notes.

FASKEN MARTINEAU

III. Case Study #1: Choosing the Right Project

FASKEN MARTINEAU

A. Background

De Beers UK Limited vs. Atos Origin IT Services UK Limited (2010)

- Example of failed system development/ implementation project, where Agile was applied to wrong project
- High Court decision:
  - Claimant: De Beers ("DB", then Diamond Trading Company ("DTC")).
  - Defendant: Atos Origin IT Services(“Atos”)
- 2006: DB decides to develop software system to support diamond supply chain management. Existing systems were out of date
- 2007
  - DB tenders software contract: Atos won.
  - Atos carries out Initiation and Analysis Phase ("IAP") to investigate & analyse business req’mts so can enter into fixed price contract.

FASKEN MARTINEAU

A. Background

- **Dec 2007: Atos dumps Agile approach and adopt Waterfall**
- Jan – Feb 2008: series of "fast track req'mts gathering workshops" are required to implement Waterfall
  - Unexpectedly generate far more req'mts = more development, & impacts technical architecture.
- Atos: cannot deliver by June (delivery date) or October 2008.
- DB. Unacceptable. Will not pay due to delays & quality of work.
- May 2008: Atos claims progress of work is delayed by
  - lack of co-operation from DB;
  - very significant increases in scope of the work
  - non-payment of fourth invoice.
- June 2008: Atos suspends all further work.

FASKEN MARTINEAU

B. Why did project fail?

- Evidence was unclear, but delay most likely caused by 4 factors:
- Atos: Lack of understanding of the business
  - **Insufficient resource commitment**: too few business analysts working on understanding DB's business (to a lesser extent, also on DB's side.)
  - **Insufficient pre-contract due diligence**: Extended post-contract time was req'd to establish DB's req'mts - complexity of DB's diamond sorting/aggregating processes was not fully appreciated at outset by Atos.

FASKEN MARTINEAU

B. Why did project fail?

- DB: Failure to communicate business requirements
  - **Lack of availability of key personnel** within DB:
    - were not attending required workshops
    - generally not available to provide info to Atos
  - **DB was still undecided** as to what kind of IT system it wanted implemented, in certain areas

C. Problem #1: Wrong Project

- Atos internal, contemporary report "**The DTC Problem**" is classic description
- Applying the Agile Project Checklist to the Project:
  - "DTC was originally intended to be developed agile-style… It became apparent that this wasn't going to work. This is for several reasons:
  - • The application is much larger than was originally thought, in terms of function points. [Not good for large-scale development.]
  - • The application is much more integrated and complex than was originally thought: a huge end-to-end multi-country workflow, with many of the same business concepts and sub-processes popping up at many points in the workflow and many "wrinkles" in the detail of the processes. [Not good for large-scale development.]
  - • The customer is demanding, particularly on the technical side, and this did not fit well with an agile approach to build." [Not good for order-based development culture.]

FASKEN MARTINEAU

C. Problem #2: Switching to Waterfall Mid-Project

Report continues:

- Accordingly the decision was taken to move towards a much more waterfall-style approach. However, this was not reflected in the team organisation, or in the definition of artefacts to be produced.
- As soon as arrived I observed that build team organisation was still divided into functional silos [i.e. iterative teams]
- we have started "specialising" so as to be able to provide the specific system support such a large and complex system requires. [i.e. moving from Agile well rounded, experienced programming role, to Waterfall specialization model]
- the workflow work … has exposed a massive gap. We have signed-off business functional req'mts but we have no definition of what system is to be built [i.e. we have no specifications.]

D. Conclusion

- Points to pitfalls of choosing wrong methodology & then trying to fix by switching mid-project
- Therefore, should ensure correct methodology is adopted in the first place
- Postscript to *Debeers*:
  - DB entitled to recover £4.4 million to cover costs of an alternative replacement system BUT had to deduct £3 million costs DB would have incurred had Atos not terminated the contract
  - nobody came out looking good

FASKEN MARTINEAU

IV. Case Study #2: *Choosing the Right Contract*

FASKEN MARTINEAU

A. Issues re Scoping

- Large agile implementation project
- High-level business case, with description of final system implemented
- Implementation Agreement executed
- System to be developed & implemented by:
  1. Releases (e.g. R1, R2, etc.), broken into
  2. Phases: i.e. Inception; Development; Stabilization; Deployment and Support)
  3. Two SOWS: (1) Inception SOW, & (2) System Delivery SOW
     - Inception SOW: generates System Delivery SOW, which delivers rest of Phases

FASKEN MARTINEAU

A. Issues re Scoping

- Where are the requirements expressed?
  - 1st: In detailed list scheduled in agreement…but then
  - 2nd: In the Inception SOW…but then
  - 3rd: In the System Delivery SOW
- Agile Rationale: vendor does not yet know what reqmts can deliver for each Release until do work in Inception SOW to generate those req'ts.
- Impact: don't have reqmts settled at:
  - signing of agmt, or
  - beginning of each Release as each is signed off

= "floating scope" issue

= customer doesn't know if will accomplish goals

FASKEN MARTINEAU

B. Issues re Payment

- If requirements are floating, what about payment?
- If accept as principle that agile development will not provide certainty as to outcomes, then payment structure should be flexible enough to line up exactly with outcomes which are achieved
- EVM, if used correctly is a good tool to achieve this

FASKEN MARTINEAU

B. Issues re Payment

1. Formulae
   a. Variables
      - **Planned Value:** valuation of planned work, based on cost budget for work completion in time period
      - **Actual Cost**: the total actual cost in time period
      - **Earned Value:** pre-defined “earning rules”/ to quantify achieved work during given period.
   b. Calculations
      - **Schedule Variance** (Earned Value minus Planned Value)
      - **Cost Variance** (Earned Value minus Actual Cost)
      - **Cost Performance Index** (Earned Value divided by Actual Cost)

FASKEN MARTINEAU

B. Issues re Payment

2. Metrics: Using Story Points

- Story points are form of EVM, linking payment to value earned
- **Right Way:** allocate to system stories based on both:
  - value to the customer, and
  - vendor work effort,
- **Wrong Way:** agmt had story points:
  - allocated solely by vendor work effort, not value earned.
  - not related to payment
  - in effect, they simply became a work estimation tool for vendor
- Rather, payment was milestone-based.
- But floating scope presents milestone challenges

FASKEN MARTINEAU

Conclusion

FASKEN MARTINEAU

Conclusion

- Successful Agile projects are dependent on two key factors:
  1. Ensuring that this is the right project for Agile.
  2. Ensuring that this is the right contract for an Agile project
- Need to ensure that:
  - neither party has “cherry-picked” only those elements of the Agile structure which best suit them
  - the implementation contract reflects actual workings of the Agile project

FASKEN MARTINEAU

Biography

John Beardwood
Partner
416.868.3490
jbeardwood@fasken.com

John Beardwood is Co-Chair of the Technology and Intellectual Property Practice Group, with his practice emphasizing on outsourcing and procurement, technology and privacy law related matters. John is regularly listed among the world's preeminent internet and e-commerce lawyers in *Who's Who Legal*, *Chambers, The Best Lawyers in Canada,* the *Legal 500*, and *Lexpert*. *Who's Who Legal - The International Who's Who of Business Lawyers* has identified John as being both one of the two most highly nominated Canadian lawyers in the guide, and one of the ten "most highly regarded individuals" globally.

FASKEN MARTINEAU