Agile Contracts

Agile Contracts

- Goal: successful project
  - Not the same as successful contract
- Foster system thinking
  - Avoid silo mentality
- Involved lawyers should understand agile principles

Agile vs. Traditional

Agile	
Time:	Fixed
Cost:	Fixed
Scope:	Variable

Traditional	
Time:	Variable
Cost:	Variable
Scope:	Fixed

Quality?

Key Pillars

- Fixed price work packages
- Early termination option
- Flexible changes
- Additional work always possible
- Ranged estimates

CONTRACT KEY POINTS

Contract key points

- Delivery cycle
- Project scope
- Change control
- Termination
- Acceptance
- Deliverables
- Liability limitation
- Warranty
- Timing of payment
- Pricing

Delivery cycles

- At the end of each timeboxed iteration deliver a deployable system with useful features
- Key elements
  - Doneness and deployability
  - Duration
  - Timeboxing

Project scope

- Agile contracts do not define an exact and unchanging project scope
  - Target-cost contract
    - Scope and details identified as best as possible
    - Mechanisms for change
  - Progressive contract
    - Scope is defined with one iteration horizon
  - Maintenance and additional work are implicitly included

Project Vision

For *<target customer>*
Who *<statement of the need or opportunity>*
The *<product name>* is a *<product category>*
That *<key benefit, compelling reason to buy>*
Unlike *<primary competitive alternative>*
Our product *<statement of primary differentiation>*

G.A. Moore,
Crossing the Chasm

Change control

- Change is at the heart of agile
- Changes
    - Relationship between parties
        - Defined in contract
    - Project scope
        - Avoid traditional method (e.g. change management board etc.)
        - Different levels of flexibility

Termination

- Termination is a unique change option
- The ideal termination model is to allow the customer to stop, without penalty, at the end of any iteration
    - Early termination should be viewed as a positive, desirable event in an agile project
    - Typically a sliding scale of penalty to the customer that reduce over time

Acceptance

- Contract should define the framework for acceptance only
- Acceptance means conformance to
    - A prior agreed-upon acceptance test suite (automated)
    - The definition of done
- Possible involvement of prospective users in the acceptance

Deliverables

- Avoid list of deliverables
  - Distracts focus from working software and waste effort in less useful activities
  - Attracts focus to conformance rather than collaboration to create useful software
  - Creates a (false) sense of control
  - Reinforces the (untrue) belief of predictability
- Documentation may be useful for maintenance, not earlier

Limitation of Liability

- Early and frequent delivery mitigate liabilities
  - Issues get discovered earlier
  - They get fixed soon
- Warranty is similar

PRICING AND PAYMENT

Contract models

- Fixed-price, fixed-scope (FPFS)
- Variable-price, variable-scope progressive
  - Capped-Price, Variable-Scope
  - Capped-Price, Partial-Fixed-Scope
  - Fixed-Price, Variable-Scope
- Shared pain/gain
  - Target cost
  - Multi-phase variable model
  - Profit sharing

Fixed-price, fixed-scope

- Optimal for:
  - Customer to know cost for budgeting
  - Supplier to book the total order value
  - Sales person to book and get commission
  - Manager to order and forget

Fixed-price, fixed-scope

- Worst for project success:
  - Especially when it comes with fixed-time
  - Typically lead to loss-loss situation
  - Contingency risk hidden fee (up to 50%)
  - Lead to technical debt accumulation
    - Offshore outsources get a “rent” from change-requests

Fixed-price, fixed-scope

- Nevertheless, they are applied
  - Upfront requirements analysis, estimation, and testing
  - Added margin to account for risk
  - Agile improves cost, productivity, quality
  - Agile let know *how bad things are* as early as possible
- Payment
  - Fixed percentage per iteration
  - Lump sum on completion

Agile Misunderstandings

- FALSE Agile makes requirements are not known or estimated before start
- E.g. Scrum initial backlog creation identifies and estimates requirements

- FALSE With Agile requirements must change
- Agile provides instruments to support change, does not mandate it

Flexibility in FPFS

- Replace existing requirements with new ones of equal effort
- Change order of implementation
  - I.e., reprioritize
- Improve “definition of done”
- Additional releases priced T&M
- Early termination
  - Pay 20% of remaining unbilled value

Variable-price, variable-scope

- Master service agreement:
  - Organizations relationships
  - Pricing scheme per iteration
    - Fixed price per iteration, T&M, etc.
- Customer exposure is limited
  - Termination possible at any iteration, with a working system
- Before each iteration customer and supplier agree on goal
  - 1 or 2 iterations ahead

Variable-price, variable-scope

- Capped-price
  - Cap upon a pricing scheme, e.g. T&M
- Non-binding release Backlog
  - Non-binding allows flexibility
  - Starting point for common vision
  - Base for estimating the iteration cap

Flexibility levels

- (Price, Scope, Time) can range from
  - Fixed to
  - Flexible
- Examples
  - Capped-Price, Partial-Fixed-Scope
    - Small subset of requirements fixed
  - Fixed-Price, Variable Scope
    - With fixed-time becomes optional scope

Pricing Schemes

- Lump sum / Fixed Price
- Time and Materials (T&M)
- Fixed price per iteration
- Fixed price per unit of work (UoW)
- Hybrid (shared pain/gain)
- Pay per use

Time and Materials

- T&M contract fit perfectly agile
- Iterative nature solves typical issues:
  - Endless cycle of payments before seeing useful results
  - Good value for the money?
- Possible exposure mitigation:
  - Capped T&M per iteration/project/release
  - Capped T&M per iteration w/adjustment

Fixed price per iteration

- Requirements defined and agreed-upon before the iteration
  - Like fixed price per project with smaller scope
  - Supplier must estimate correctly
  - Supplier adds a contingency fee for risk
- Highly flexible or no predefined requirements
  - Customer trust is key
  - Transparency, frequent delivery

Fixed price per unit of work

- UoW == running, tested features
  - Working software is the primary measure of progress
  - Measure of features
    - Story points
    - Feature points
    - Use Case points
    - Function points

Fixed prices per UoW – Issues

- Feature measurement
  - Clear and shared framework
  - Possibly standardized measure
  - Possibly independently certifiable
    - E.g. Function points
- Business value vs. size points
  - Loose correlation
  - Impact estimation

Gilb, T., 2005. Competitive Engineering, Butterworth-Heinemann

Determining UoW price

- Average based on previous projects
- Customized amount
  - A few iterations w/other pricing scheme
    - Average price per UoW
    - T&M
  - Then agreement on fixed price per point

Risk shifting and sharing

- Aligning motivations of both parties
- Sharing
  - Both parties have skin in the game
- Shifting
  - Risk on the party that is accountable
  - Requirements-related risks in the hands of customer (what)
  - Technical-related risks in the hands of the supplier (how)

Target cost

- Cost definition
  - Identify and estimate requirements and change costs (realistic)
  - Supplier establish the target cost and calculate target profit (e.g. 15%)
  - Share result with customer
- Project execution
  - Actual cost tracking and sharing
  - Determine adjustment
    = (ActualCost – TgtCost) * CustomerShare
  - Payment = TgtCost + TgtProfit + Adjust

Target cost

- Key elements
  - High transparency and near-real-time, open-book project accounting
  - Collaboration for continuous improvement
  - Agreement on guidelines for target-cost adjustment
    - Inherently fuzzy
  - Test-driven acceptance

Hybrid shared pain/gain

- Price is composed of
  - Price for effort (discounted)
  - Price per unit of work

Hybrid

- Fictional case

Estimate	1800	FP
Team	6	FTE
Iteration	2	Weeks

Productivity: 2.5 FP/PD
Cost: 500 €/PD

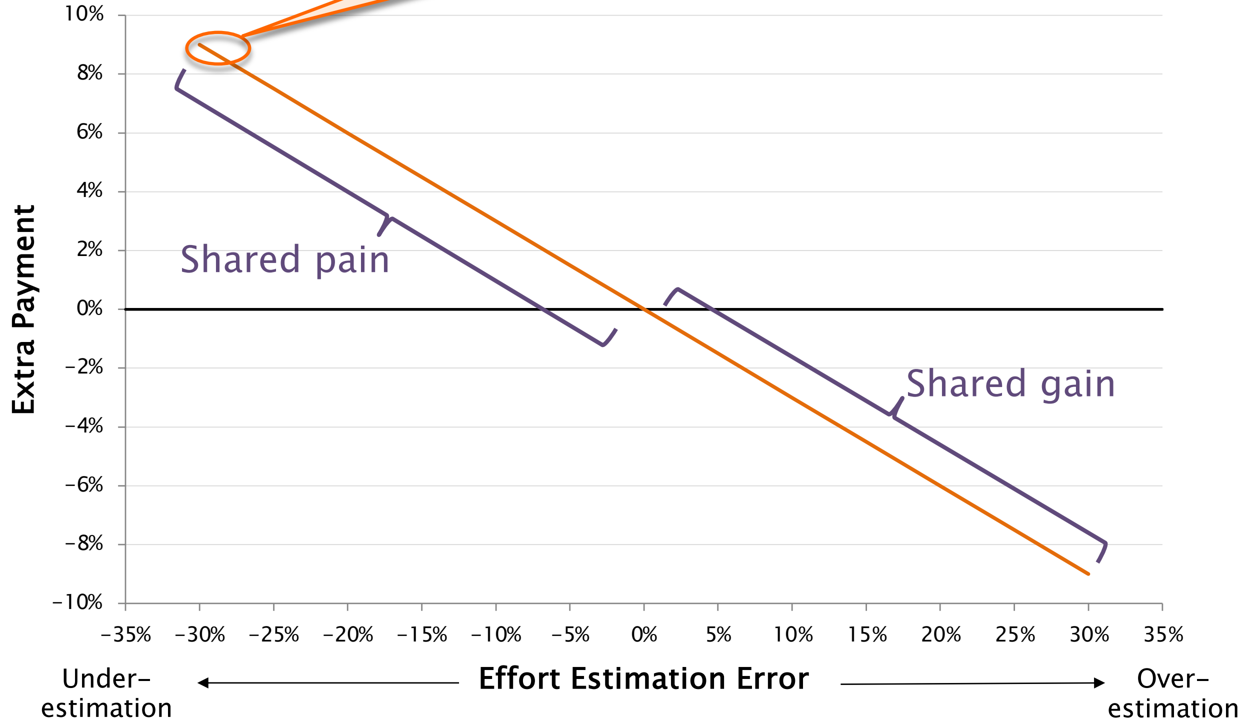
Velocity	150	FP/Iteration
Duration	12	Iterations
Effort	720	PD
Cost	360 000	€

- Price
  - 150 €/PD (discounted: 70%)
  - 140 €/FP

Computed to get the PD cost in case of correct estimation

Hybrid

70% of extra cost/saving is charged on supplier



Multi-phase variable model

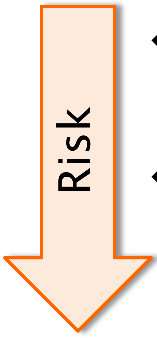
- Projects have a changing risk profile over time
  - Hopefully with an improving trend
- Multiphase adjust contract type to accommodate such change

Contract evolution (example)

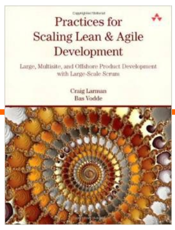
- FPFS per project
  - Agile exposure is attractive
- Progressive, T&M with bonus/penalty
  - Penalty may trigger gaming
- Progressive, fixed price per unit of work
- Progressive, capped T&M
  - Support and maintenance

Risk bounding

- Level of trust is a key element
- E.g. with a new supplier:
  - One year-long, fixed-price, fixed-date, variable-scope
  - Series of two-months, fixed-price, fixed-date, variable-scope contracts with termination option at each iteration
- A few low-risk contract cycles may build trust



References



- Craig Larman and Bas Vodde. “Practices for Scaling Lean & Agile Development: Large, Multisite, & Offshore Product Development with Large-Scale Scrum” Addison-Wesley
- http://www.agilecontracts.org
- https://www.toptal.com/agile/software-costs-estimation-in-agile-project-management