THE EVOLUTIONARY CONTRACT MODEL

The Agile Business Conference
6 October 2011

Susan Atkinson
Legal Director
gallenalliance Solicitors
CHARACTERISTICS OF DEVELOPMENT

• Development of products/services involves the optimisation of knowledge creation

• It has been proven that the amount of knowledge created is optimised with feedback:
  - Feedback makes learning faster and more efficient
  - Feedback cuts through a ‘noisy’ environment
  - Feedback reduces the cost of failure associated with risk-taking
  - Fast feedback gives a sense of control

• Development is a complex process subject to much change
EMPIRICAL PROCESS CONTROL

- Clear Business Goal
- Adaptation
- Inspection
- Visibility

- Time split into timeboxes
- Solution split into increments
- Metrics on the development process
- Inspections frequent enough to detect unacceptable variances
- Adjustment to be made quickly to minimise further deviation
- Quantified and measurable business outcomes
SIGNIFICANCE OF EMPIRICAL PROCESS CONTROL

Å Results-focused:

- Measuring completed increments of the solution
- Not measuring tasks or activities
- Work in progress has no value

Å The solution is not defined:

- Pre-defined requirements and specifications amount to amateurish design
- Instead the business goal is defined

Å Contractual commitment of supplier to ‘Complete’ Requirements from the PRL with an agreed amount of project effort – to enable adaptability (NB. Alternative approach – commitment re goal)
## COMPARISON OF PROCESS CONTROLS

<table>
<thead>
<tr>
<th>Defined Process Control</th>
<th>Empirical Process Control</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Control mechanism</strong> - defined plans i.e. the requirements and milestones</td>
<td><strong>Control mechanism</strong> - feedback – requirements: (i) clear goal, (ii) visibility, (iii) inspection, (iv) adaptation</td>
</tr>
<tr>
<td><strong>Success</strong> - conformance with defined plans e.g. does the design satisfy the requirements specified upfront?</td>
<td><strong>Success</strong> - realisation of quantified desired business outcomes that deliver value to the customer</td>
</tr>
<tr>
<td>Simple, predictable processes – this is the control process of choice</td>
<td>Simple, predictable processes – not applicable</td>
</tr>
<tr>
<td>Complex, unpredictable processes – not appropriate:</td>
<td>Complex, unpredictable processes – this is the control process of choice - the customer has frequent opportunities to inspect and adapt i.e. to provide feedback:</td>
</tr>
<tr>
<td>➢ prescriptive approach with limited flexibility</td>
<td>➢ increases learning</td>
</tr>
<tr>
<td>➢ the plans can quickly become obsolete</td>
<td>➢ cuts through a ‘noisy’ environment</td>
</tr>
<tr>
<td>➢ conformance with the plans does not necessarily deliver value</td>
<td>➢ reduces the cost of failure</td>
</tr>
<tr>
<td>➢ it is not until testing (i.e. at the end of the project) that the customer has a sense of whether the solution delivers value</td>
<td>➢ gives a sense of control</td>
</tr>
</tbody>
</table>
STRUCTURE OF EVOLUTIONARY CONTRACT MODEL

Prioritised Requirements List

Increment 1
Increment 2
Increment 3
Increment 4

Development Timeboxes

Prioritised Requirements List

Entry into contract
Contractual gatepost

SOW

Development?

Deployment?

Deployment?

Deployment?

FEASIBILITY

FOUNDATIONS

EXPLORATION AND ENGINEERING
ENTRY INTO CONTRACT - KEY SCHEDULES

- **Vision Statement** – ASPIRATIONAL. What you’re doing, why you’re doing it and how you’ll know if you’re successful e.g. ‘to put a man on the moon’

- **Desired Business Outcomes** – ASPIRATIONAL. High level objectives of the solution from a business, operational and technical perspective which deliver value to the customer – ultimately to increase profits or to reduce spend

- **Constraints** – BINARY. These may be:
  - scope constraints e.g. compliance with regulations
  - schedule constraints e.g. a key completion date
  - budget constraints
FOUNDATIONS – KEY DELIVERABLES

- **Business Foundations** – expands upon the desired business outcomes with a quantifiable and measurable scale

- **Solution Foundations** – an outline plan for the technical and business architecture/design of the solution

- **Delivery Plan** – segments the solution into the smallest possible solution increments that realise the desired outcomes and sets out the order in which they are to be delivered

- **The Prioritised Requirements List (PRL)** – an evolving prioritised list of all requirements (items of work) which may be relevant to the solution

- **Management Foundations** – describes essential governance and organisation aspects of the project and how the project will be managed

- **Delivery Control Pack** – reports, documents and logs relating to the ongoing status of the project

- **NOTE** – also calibration of the team
THE PRIORITISED REQUIREMENTS LIST (PRL)

- Includes all requirements (items of work) relevant to the solution
- May not be a document
- Requirements are prioritised
- Dynamic
- Evolves
- **Must** be within the scope of the contract
- The customer representative **only** controls the PRL
STRUCTURE OF EVOLUTIONARY CONTRACT MODEL

Prioritised Requirements List

- Increment 1
- Increment 2
- Increment 3
- Increment 4

Development Timeboxes

- SOW
- SOW
- SOW
- SOW

Entry into contract

Contractual gatepost

Deployment?

Deployment?

Deployment?

Deployment?

FEASIBILITY

FOUNDATIONS

EXPLORATION AND ENGINEERING

gallenalliance
CONTRACTUAL OBLIGATIONS

SOWs: Contractual commitment:

- to ‘complete’ Requirements comprising the Minimum Useable Subset (MUS) with an agreed amount of project effort by the Completion Date OR

- to realise an agreed percentage of the desired business outcomes by the Completion Date

All Requirements to be estimated in terms of amount of project effort involved for them to be ‘Completed’

Requirements on the PRL can be exchanged provided that they are replaced with Requirements that are estimated to involve the same project effort
COMPLETION OF REQUIREMENTS

Definition of ‘Completed’:

- Software and deliverables are within the parameters of the constraints and meet the acceptance criteria

- Software: at a minimum, working software, integrated with all previous solution increments and ready to be deployed to a production-ready environment

- Completion has been confirmed by the customer representative

Warranty that the Solution Increment meets the criteria for ‘Completion’ for an agreed period of time following the Completion Date
Desired business outcomes. A scale against which realisation of the desired business outcomes can be measured

Non-functional attributes. Examples include: availability, adaptability, usability, workload capacity

Efficiency of the project. Relevant metrics may include:
- Velocity – total units of work completed in a timebox
- Burndown / Burnup charts – chart which plots number of units of work that have been completed against number of units of work to be completed

Maturity of the agile practices. Examples include:
- Build management and continuous integration – extent to which process that builds the application and runs the tests upon every change is automated
- Testing – extent to which various tests e.g. unit testing and regression testing are automated
- Release management – the definition, support and enforcement of processes for preparing software for deployment to a live environment
## COMPARISON OF CONTRACT MODELS

<table>
<thead>
<tr>
<th>Traditional Contract Model</th>
<th>Evolutionary Contract Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>The requirements are contractual and specified upfront.</td>
<td>All requirements relevant to the solution are added to the PRL which is not contractual and which evolves.</td>
</tr>
<tr>
<td>Changes 'controlled' by means of the change control mechanism.</td>
<td>Change is accommodated within the non-contractual PRL.</td>
</tr>
<tr>
<td>Analysis, design, development and testing occur sequentially.</td>
<td>Concurrent design and development.</td>
</tr>
<tr>
<td>An all-or-nothing solution.</td>
<td>The solution is broken down into solution increments.</td>
</tr>
<tr>
<td>Constituent 'modules' of software are worked on independently until integration takes place.</td>
<td>A continuous working and stable software system.</td>
</tr>
<tr>
<td>Testing is used as a contractual tool.</td>
<td>Testing performs many roles and is an integral part of the development process.</td>
</tr>
<tr>
<td>Success is measured by reference to conformance with the plans.</td>
<td>Success is measured by reference to the extent to which the completed solution increments realise the desired business outcomes and therefore deliver value.</td>
</tr>
</tbody>
</table>
# COMPARISON OF RISK PROFILES

<table>
<thead>
<tr>
<th>Traditional Contract Model</th>
<th>Evolutionary Contract Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer is committed financially to <em>entire</em> project.</td>
<td>Financial commitment initially to Foundations only and then on an Increment by increment basis.</td>
</tr>
<tr>
<td>No value delivered until the entire project (i.e. <em>all</em> of the requirements) has been completed.</td>
<td>Value delivered at the end of <em>every timebox</em>.</td>
</tr>
<tr>
<td>No end-to-end system created until the <em>end</em> of the development project i.e. integration doesn’t happen until just before testing.</td>
<td>End-to-end and fully working system (albeit only a solution increment) delivered at end of <em>first timebox</em>, and this continues to grow with <em>each timebox</em>.</td>
</tr>
<tr>
<td>If the project is going awry the customer generally won’t know UNTIL testing at the <em>end</em> of the project.</td>
<td>If the project is going awry, the customer should detect it during the <em>Foundations</em> or at the end of <em>every timebox</em>.</td>
</tr>
<tr>
<td>There is <em>no</em> attempt to control the order in which the requirements are tackled.</td>
<td>The highest risk and highest value Requirements are tackled first. Low value Requirements are ‘dropped’.</td>
</tr>
</tbody>
</table>
CHARGING MODELS

- **Foundations** – fixed price or time & materials

- **Exploration and Engineering** – options in respect of each Increment:
  - Time & materials
    
    CAUTION: budget not fixed and little incentive for supplier to be efficient
  
  - Fixed price (i) per agreed unit of effort or (ii) per Increment (i.e. agreed aggregate units of effort for all requirements batched in the Increment)
    
    CAUTION: less collaborative; reduces opportunity for continuous improvement
  
  - Incentive for delivering specified metric of desired outcomes on or before Completion Date
  
  - Incentive for increased maturity of agile practices
  
  - Incentive – agreed units of work delivered before Completion Date or more than agreed units of work delivered by Completion Date
    
    CAUTION: may incentivise more of the wrong code; may increase technical debt
THANK YOU

Susan Atkinson
gallenalliance Solicitors
9 Devonshire Square
London EC2M 4YF
United Kingdom
Tel: +44 20 7084 6392
Email: satkinson@gallenalliance.com