IBM

IT Specialist Institute 2007

Successfully implementing WebSphere Portal integration and development projects using an iterative method

Richard Shooter
Peter Greaves

Innovate….Integrate….Deliver

Challenges Facing (Portal) Integration Projects

- **Integrating services (products, interfaces, SOA, …) introduces a dependency on the service being integrated to/with**
  - The integration point (Portal) takes the hit for any delay
- **Integration may require custom development**
  - The solution may be novel, estimates are difficult, higher risk
- **Integration into the enterprise architecture (database, directory)...**
  - …requires that such an architecture already exists
    - Does it exist?
    - Is it robust and scalable enough to support an enterprise-wide presentation architecture?
    - If not, can it be made so under this project? If not, what are the compromises?
- **Integration is disruptive**
  - Enables (requires) previously unconnected groups to co-operate
  - (Project) Funding models may no longer be applicable or appropriate
- **Customers may require greater involvement of their own staff: "doing it WITH rather than TO"**
  - Estimates are more difficult: skill levels, availability, priorities, method
- **Clearly not unique to Portal projects!**

What factors make portal projects high risk?

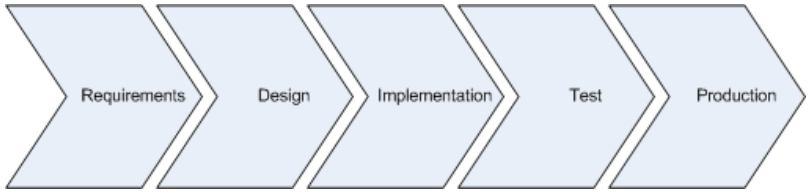
- **Portal projects are SOAs : strategic, not tactical**
  - Provides "…core portal services that aggregate applications and content and deliver them as role-based applications"
    - Highly invasive : how often do our clients implement a SOA?
- **Implications of technical complexity and novelty**
  - Our products (WP/WAS, Collaboration tools, IM, Tivoli etc) are non-trivial
    - Portal is a multi-tier framework application, requiring skills in many areas
  - Client environments (directories, database, analytics, deployment environments, existing application landscape) are combinatorially unique
  - Our delivery organisation is complex and disconnected
- **The client's readiness**
  - Are requirements adequately concrete for decisions to be made?
  - Is the organisation fully ready and mobilised for a Portal SOA?
  - Does a governance model exist for the portal that incorporates all important stakeholders?
  - Is a strategy in-plan for incorporating existing applications? (If you build it, will they come?)

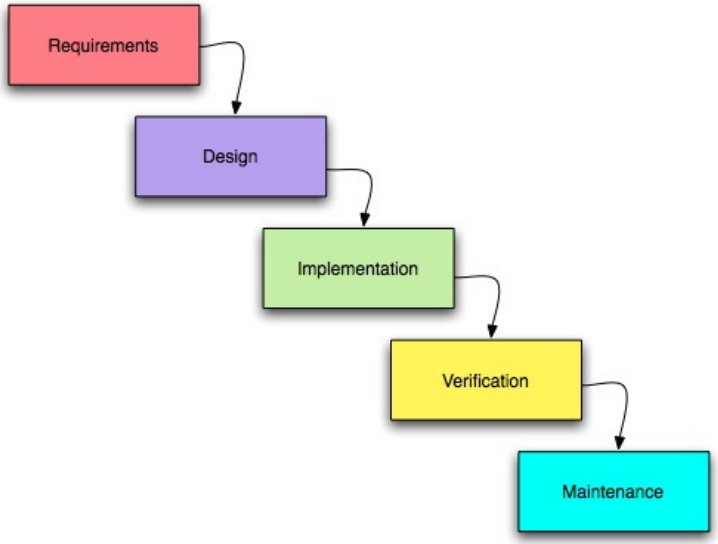
Project Approaches

- **Review of different methods to managing portal projects**
  - Single iteration or “Traditional” waterfall
  - Multiple iteration waterfall
- **Factors that influence choice of methods**
  - Our consulting approach and capabilities
  - The client’s culture and preferences
    - Is the portal project being used to drive change?
  - The project’s risk factors

The waterfall approach to delivery

- **"Traditional" project method, AKA BDUF (Big Design Up Front)**
- **Notionally each stage is completed before starting the next**
  - There is always some acceptance of iteration and feedback regardless of method adopted
- **All requirements are delivered simultaneously at the end of the cycle**
- **A sequence of disciplines and activities related to those disciplines**
- **Each phase is 100% complete and absolutely correct before proceeding to the next phase**
- **The portal's design is perfected before implementation starts**
- **Works well in projects of low risk and complexity**

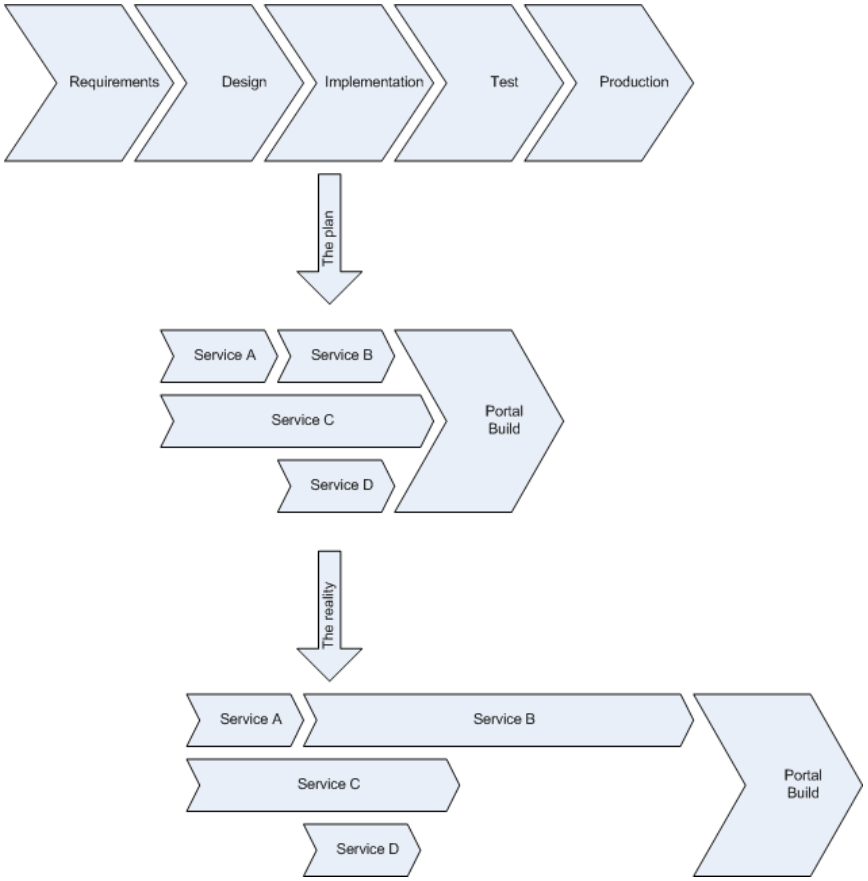




Single iteration waterfall

- **Advantages**
  - ✓ "Measure twice, cut once"
  - ✓ Clearly delineates stages
  - ✓ Simple and general purpose
  - ✓ Works well in other engineering disciplines
- **Disadvantages**
  - × Requires everything to be known in advance
    - × And not require revision in the light of subsequent knowledge (next-phase feedback)
  - × Ignores external factors and business change during project lifecycle
  - × Ignores the short (and generally unhappy) record of software project implementation
  - × Hard to "fail early"
    - × little opportunity for feedback early in project
  - × Makes estimation of project cost difficult
    - × E.g. cost of tackling new requirements in implementation phase
  - × Lacks specific disciplines that deal with software projects
    - × Not easily extensible

Slippage occurs



Slippage is likely as:

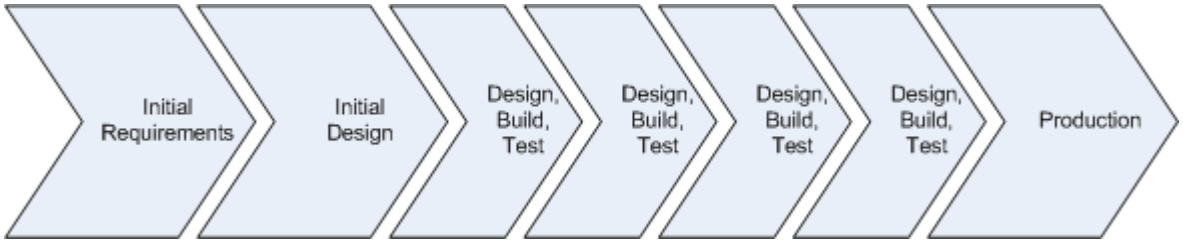
- **Services to be integrated are delayed**
- **The technology does not work as expected or understood**
  - Service(s)
  - Custom development or configuration
  - Integration of Portal
- **Requirements change or are clarified**
- **Staff become unavailable or are less productive than expected**

Waterfall Method Effects

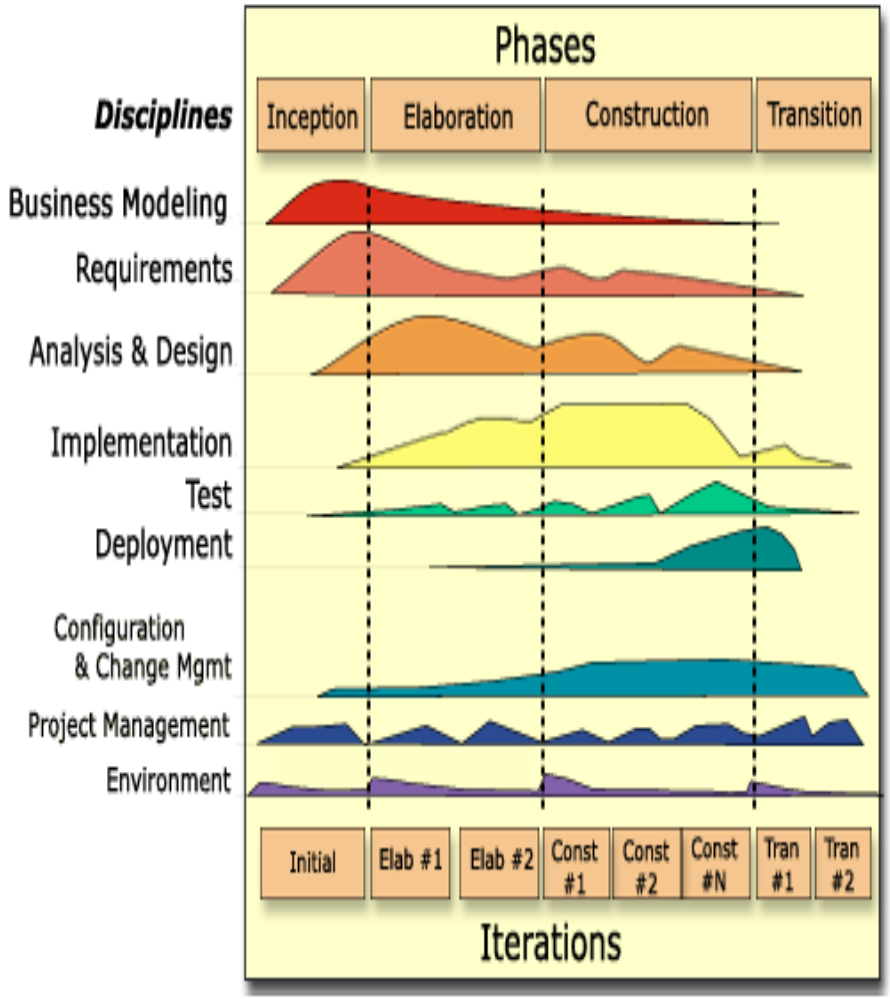
- **For schedule-sensitive projects…**
  - Testing may be squeezed to meet original deadline
    - Go live with increased risk?
    - Delay go live?
  - Or a compromise in implementation quality
    - Customer generally has commitments to scope
      - And only the implementation phase has “wiggle room”
- **No proven solution is available until end of testing phase**
  - Higher probability of “failing late”
  - Confrontational ethos pervades the project
- **Method is inflexible**
  - No opportunity to go live with subset of functionality: nothing has been tested until everything is available

Alternative Approach – Iterative Method

- **Borrow from (software) development practice**
- **Accept limits of requirements, product understanding, scheduling accuracy**
- **Structure the project as repeated Design,Build,Test cycles – prove the implementation**
- **Incrementally add functionality to each cycle**



Multiple iteration waterfall – Rational Unified Process



- **Breaks project into phases**
  - Inception
  - Elaboration
  - Construction
  - Transition
- **Phases contain iterations**
- **Iteration feedback built in**
- **6 core principles**
  - Adapt the process
  - Balance stakeholder priorities
  - Collaborate across teams
  - Demonstrate value iteratively
  - Raise the level of abstraction
  - Focus on quality

Defining the contents of the phases

- **Functionality selected by**
  - Architectural risk – early in cycle
    - Integration with external systems, novelty of technology
  - Business priorities – early in cycle
    - Key cases of use and Qualities/Constraints
  - Technical dependencies – key technologies proved early
  - Risk – bring risk forward in the project
- **Some level of parallel development may be possible**
  - Concurrency in each iteration towards coherence at the iteration end deliverable

Advantages of the iterative approach

- **Advantages for portal projects**
  - ✓Focuses on mitigation of risks and change control
  - ✓Includes formal and traceable requirements management
  - ✓Clear and unambiguous communication
  - ✓Central concern on strong architecture (architectures that work properly under "stress")
  - ✓Tackles problem of “Overwhelming complexity”
  - ✓Detects inconsistencies in requirements, designs, and implementations
  - ✓Elevates testing to peer status with other disciplines
  - ✓Requires an objective assessment of project status
  - ✓Delivers value early
    - ✓Each iteration must end with a coherent system that has clear and unambiguous value to the client
- **Disadvantages**
  - ×Higher ceremony than other methods
  - ×Adapting process for an organisation is not trivial

Summary

- **Adopting an iterative approach can de-risk integration projects**
- **The cost (ceremony) is worth the effort**
- **Clients get to change control the development of the system**
  - By varying scope, schedule or cost (not quality)
  - Within a process that facilities change throughout the system

Thank you

- **Questions?**