



MSc in Official Statistics Statistical Computing: Project Specification and Management

Andrew Westlake

Survey & Statistical Computing

63 Ridge Road, London N8 9NP, UK

+44 (0) 20 8374 4723

AJW@SaSC.co.uk (E-Mail)

www.SaSC.co.uk

What is the Problem?

- Developing any software system is difficult
 - » Users do not know how to specify what they want
 - » Users do not really know what they want
 - » Development based on fixed sequential stages is bound to fail
- Incremental development
 - » Allow for changes and refinement of requirements
 - » Feature-oriented design - don't need everything at once
 - » Risk management
- User-centred design
 - » How do IT staff find out what is needed?
 - » How do users find out what they are getting?

Management methodologies

- Gantt and Pert charts
 - » Good for tracking progress in well-defined projects with lots of interdependent tasks
 - » Essential that most of the tasks are known in advance (at design stage)
 - Can cope with delay, but not with redefinition/iteration
- SSADM and Prince
 - » Ensures that software produced is documented in detail
 - » **Not** that the users get the system they really want
- Methodology needs to recognise the need for change and to support iteration and refinement
 - » These are standard in modelling, so base on modelling process

Modelling Methodology

- UML - Unified Modeling Language
 - » OMG standard - Subsumes all previous proposals
 - » Covers all aspects, from requirements to code
 - » Focussed on software development, but can cover other processes
 - » Extensible, using Profiles, e.g. SysML, Systems Modeling Language
- Handles Complexity through Decomposition
 - » Can focus on a particular area of the model, and/or
 - » Can restrict attention to a particular level of the model
- Recognises need for refinement
 - » Different levels of detail for different purposes
 - » Over time as understanding improves
 - Users refine requirements in response to model and early implementations
 - Requirements linked to model to show dependencies and the impact of change
 - » Requirements become features of the implemented system
 - Yield scenarios for testing

Design and Development Methodology

- Rational Unified Process
 - » Based on UML in Rational Rose (now IBM)
 - » Includes requirements database
- Generic components (Coad)
 - » Prototypes of useful Class types
 - » Enrich UML by recognising common situations
- Feature-Driven Development (Coad)
 - » Methodology for managing incremental development
- Agile Model-Driven Development (Ambler)
- Maintain user involvement during the development

User-centred design

- Contextual Design (Holzblatt)
 - » Observe what users do
 - » Analyse what they did
 - » Detailed methodology - resource-intensive
- The user-centred dilemma
 - » Users report what they think is important
 - » Watching what they do can reveal hidden structure
 - » What they do may not be the best solution
- User-centred design
 - » Design operations for ease of use, not aesthetics
 - » Useful and Usable

Requirements in Design

- User needs - what are they trying to achieve
 - » Objectives and Functionality
 - » Ask Users
 - » Ask Experts
 - » Different groups have different needs
- Design Components - what is to be provided
 - » More conceptual analysis of functionality
 - » Review with Users
- Solution Components - how will it be done
- Implementation
 - » Continual review with users and designers
- See example from LATS statistical database specification
 - » in Background

User Needs - Discovery

| User Needs | Importance (1 - 3, 0 = not provided) | | |
|---|---|--------|--------|
| | Specialists | Policy | Others |
| Discovery | | | |
| Discover resources using general terminology and common names | 1 | 3 | 3 |
| Discover resources using technical terminology and specialised references | 3 | 1 | 1 |
| Discover resources by following links from one item of information to another | 2 | 3 | 3 |
| Review definitions and other background information related to figures | 3 | 1 | 2 |

User Needs - Display

| Display | Specialists | Policy | Others |
|--|-------------|--------|--------|
| View summary information as figures, and focus in on details of interest | 2 | 1 | 3 |
| View summary information as charts and diagrams | 1 | 3 | 2 |
| Read analyses and conclusions based on the data, produced by others | 2 | 3 | 2 |
| Integrated interface to different types of information | 1 | 3 | 3 |
| Presentation of information personalised to experience and preferences of user | 1 | 3 | 3 |
| Choose the level of interaction with the complexity of the system, from standard or base views and extracts, down to the full internal structure | 3 | 1 | 2 |
| View the precision or confidence associated with any displayed information | 2 | 2 | 2 |
| Set bookmarks, to ease return to information that has been found | 2 | 3 | 3 |



User Needs - Session Management

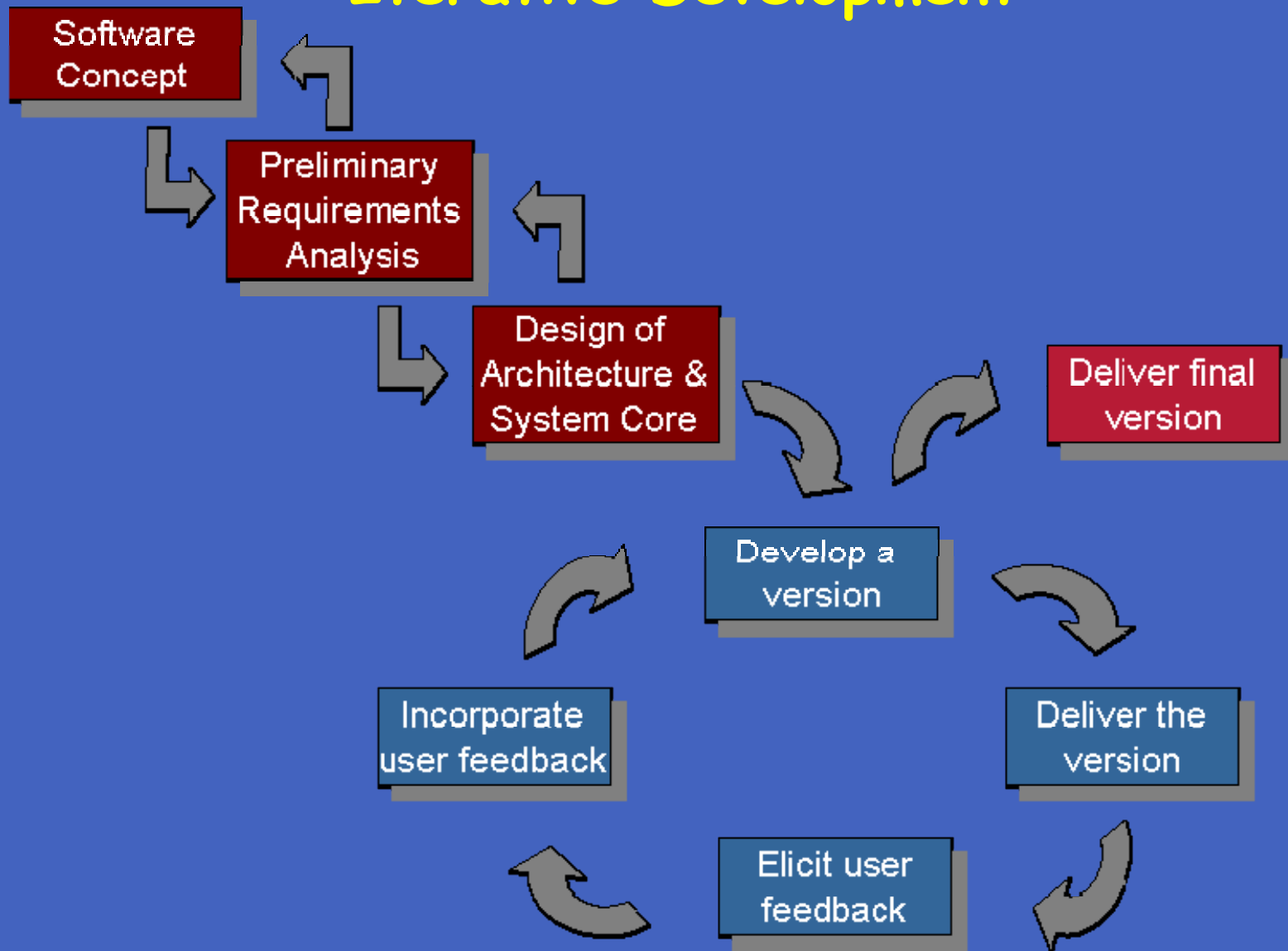
| Session Management | Specialists | Policy | Others |
|--|-------------|--------|--------|
| Break off in the middle of exploration or derivation and return to the same place at the next session | 2 | 2 | 3 |
| Review the steps taken to reach a particular selection or view or arrangement of information | 2 | 1 | 3 |
| Rerun the steps taken to reach information, when any of the input components have been changed | 3 | 1 | 2 |
| Where source information is updated, be able to refer explicitly to particular older versions (to maintain consistency with extracted information) | 3 | 1 | 2 |

Design Components

| | |
|-------------------------------------|---|
| Discovery | Catalogues using standard terminology will be aimed at subject specialists, a thesaurus component using common terms and alternatives will support non-specialists, and a general free text search facility over the descriptions, labels and analysis content should be useful to all. |
| Bookmarks and Links | A general bookmark facility will allow users to remember where useful information is found, and will also support the construction of subject-specific indexes and catalogues aimed at particular groups of user. This should be essentially the same mechanism as is used for links between metadata and the referenced objects in the database. |
| History | A general history (or audit trail) facility will keep track of operations performed by users, allowing them to be reviewed or re-run. |
| User Interface and Resources | Users should experience default settings based on membership of user groups, but be able to set their own preferences for various components of the interface. Private user storage is needed for bookmarks, histories, versions of summaries and analyses, and for returning to incomplete investigations. Ideally we need a workbench approach to the interface. |

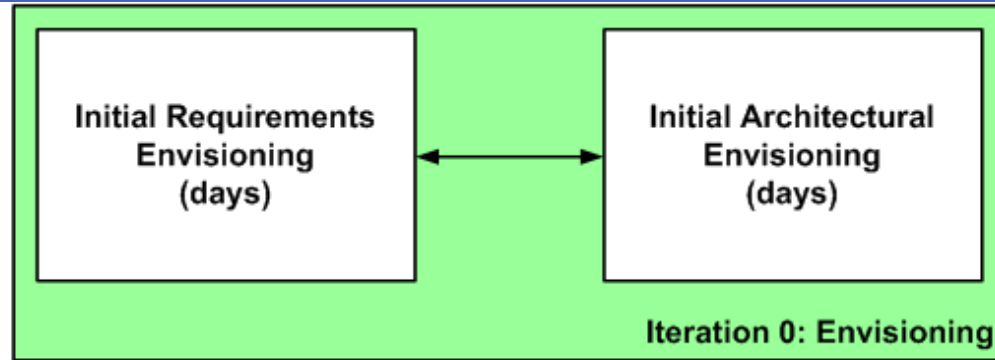


Iterative Development

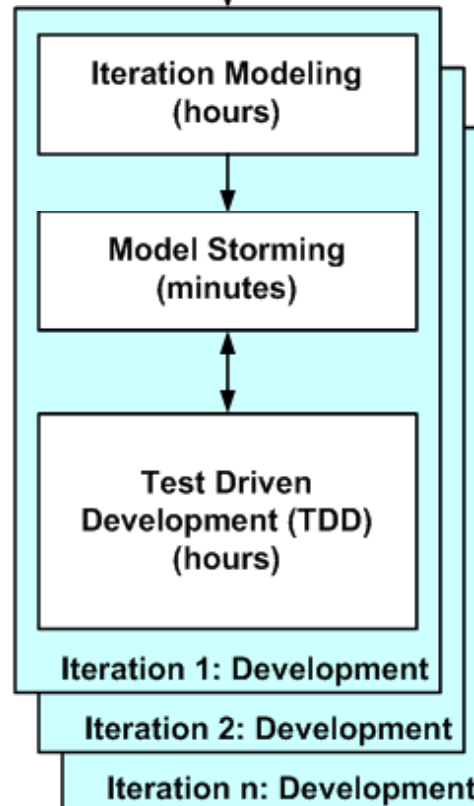


Agile Model-Driven Development

- Identify the high-level scope
- Identify initial "requirements stack"
- Identify an architectural vision



- Modeling is part of iteration planning effort
- Need to model enough to give good estimates
- Need to plan the work for the iteration
- Work through specific issues on a JIT manner
- Stakeholders actively participate
- Requirements evolve throughout project
- Model just enough for now, you can always come back later
- Develop working software via a test-first approach
- Details captured in the form of executable specifications



Conclusions

- Be involved in the design and development process
- Understand modelling
 - » Be proactive in the requirements and top-level stages
 - » Follow what is being developed and be critical
- Press for a well-defined development methodology
- Insist on iterative development, with frequent feedback
- Maintain user involvement during the development