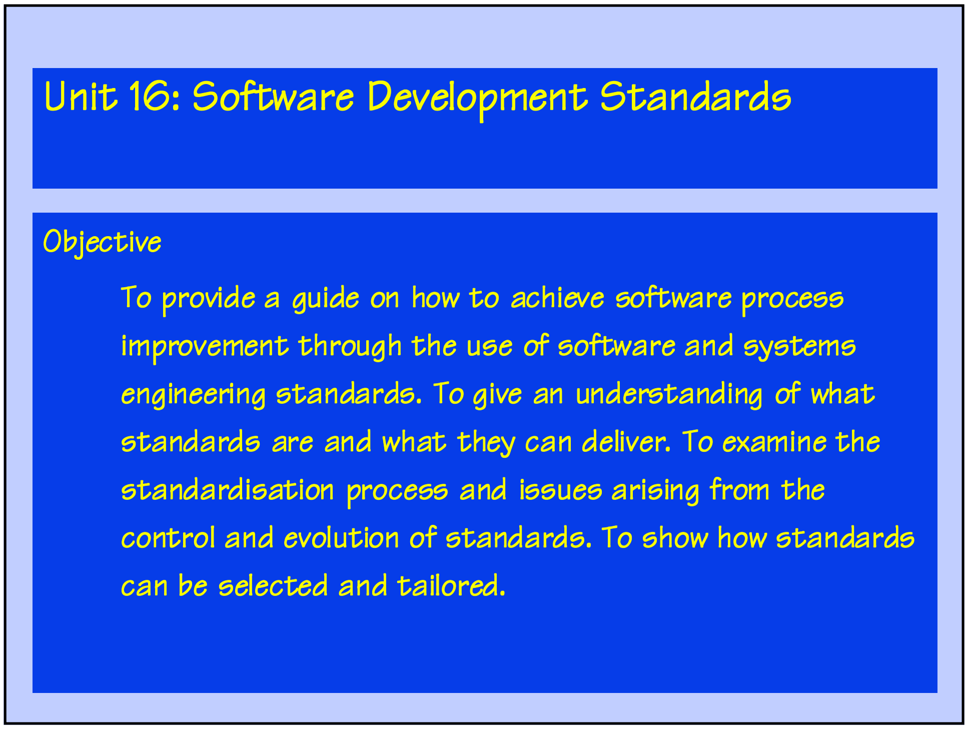
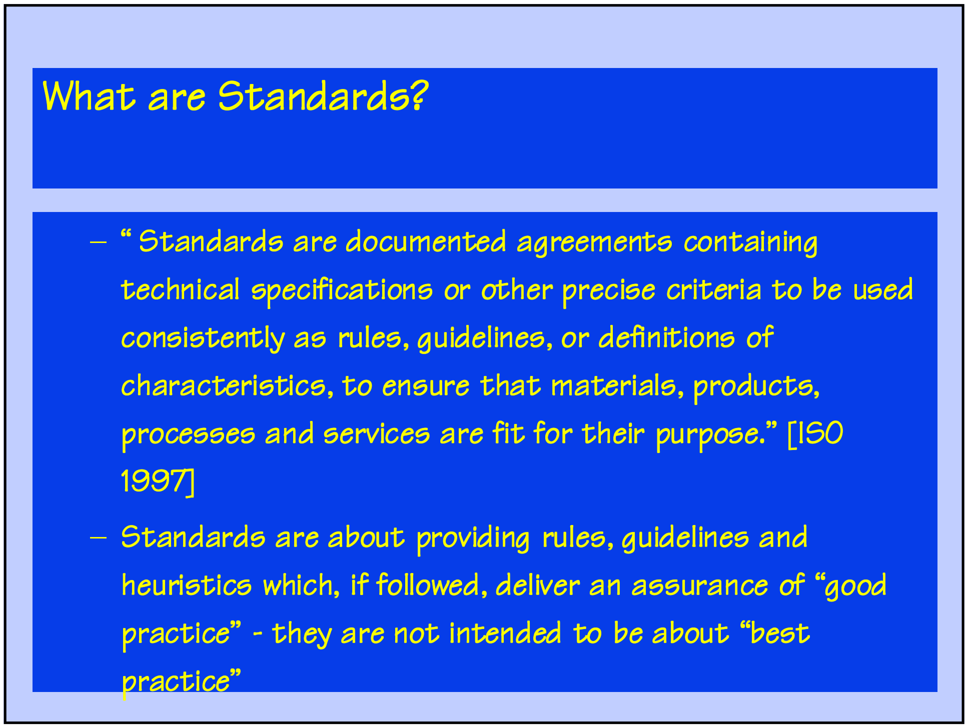


Unit 16: Software Development Standards

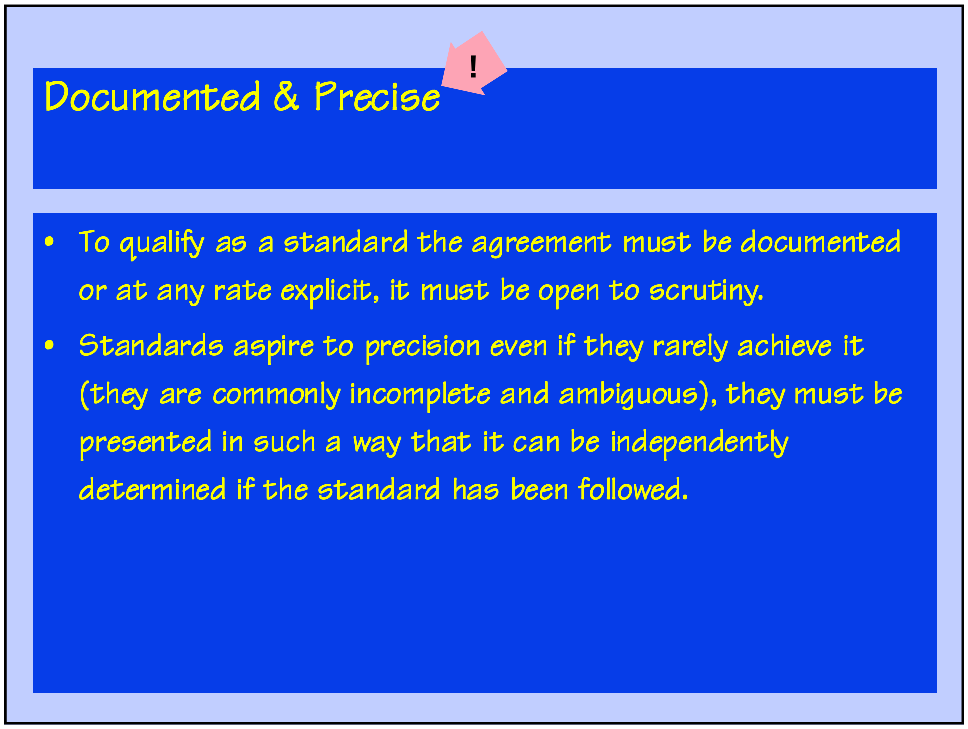
Objective

To provide a guide on how to achieve software process improvement through the use of software and systems engineering standards. To give an understanding of what standards are and what they can deliver. To examine the standardisation process and issues arising from the control and evolution of standards. To show how standards can be selected and tailored.



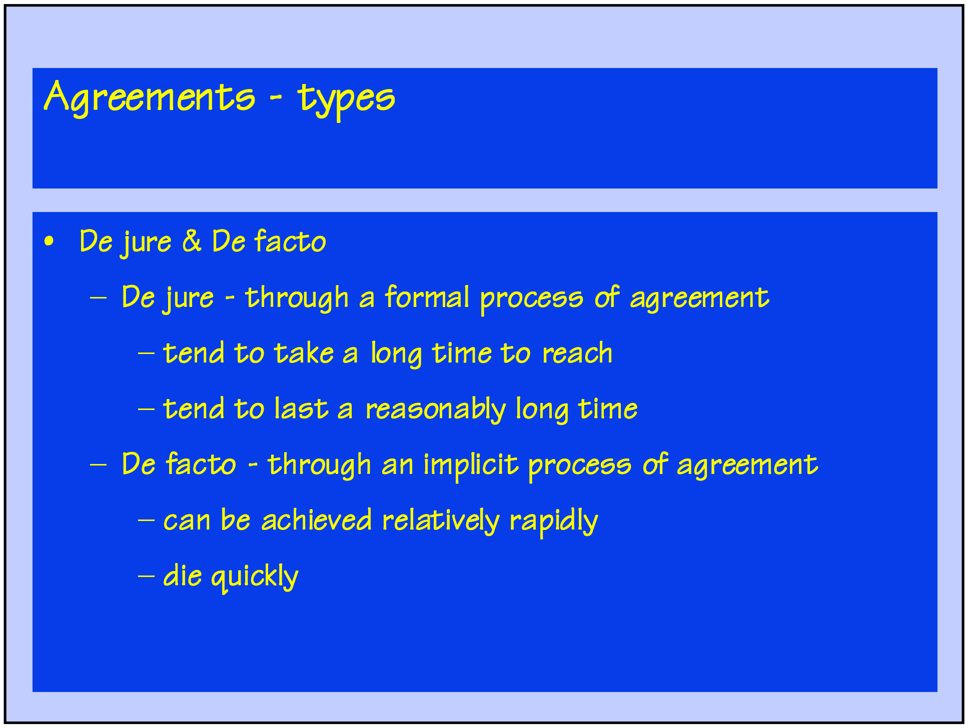
What are Standards?

- “ Standards are documented agreements containing technical specifications or other precise criteria to be used consistently as rules, guidelines, or definitions of characteristics, to ensure that materials, products, processes and services are fit for their purpose.” [ISO 1997]
- Standards are about providing rules, guidelines and heuristics which, if followed, deliver an assurance of “good practice” - they are not intended to be about “best practice”



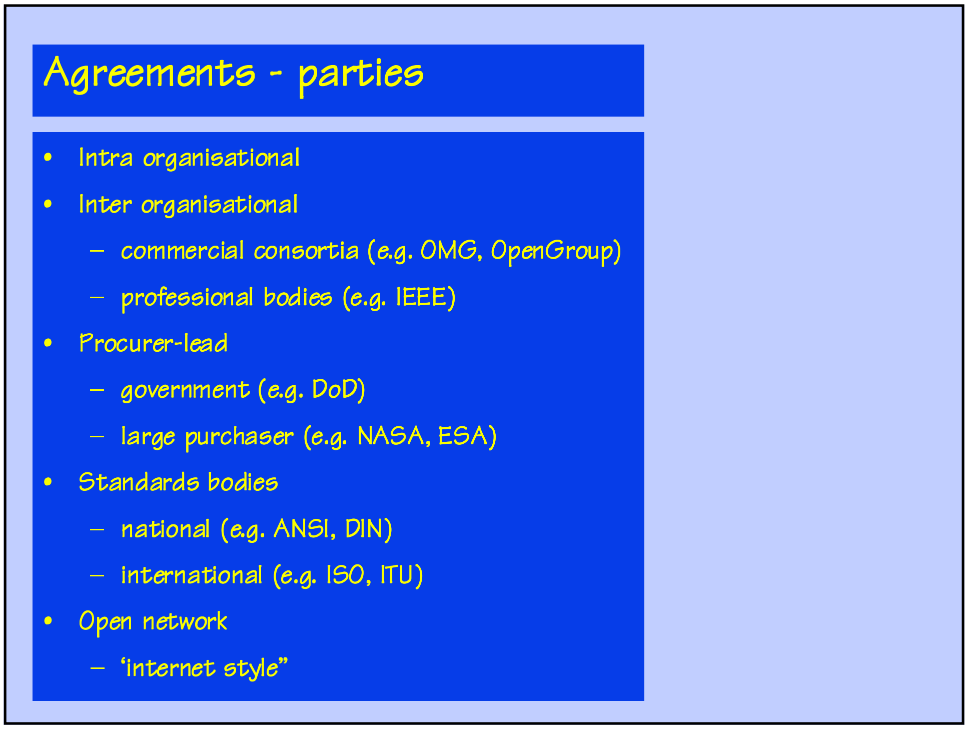
Documented & Precise

- To qualify as a standard the agreement must be documented or at any rate explicit, it must be open to scrutiny.
- Standards aspire to precision even if they rarely achieve it (they are commonly incomplete and ambiguous), they must be presented in such a way that it can be independently determined if the standard has been followed.



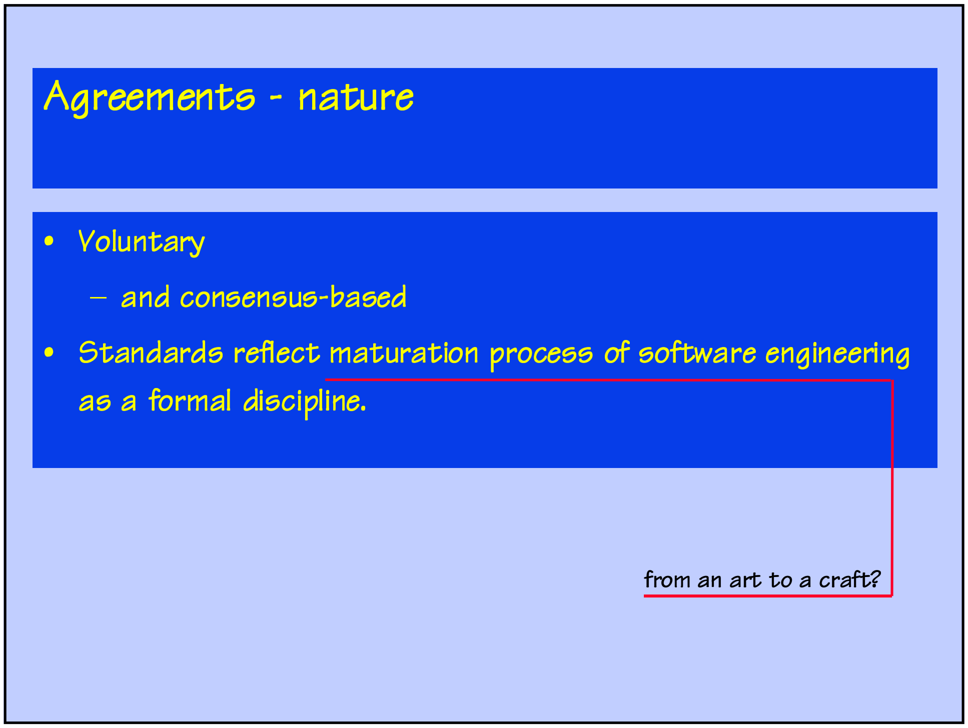
Agreements - types

- De jure & De facto
  - De jure - through a formal process of agreement
    - tend to take a long time to reach
    - tend to last a reasonably long time
  - De facto - through an implicit process of agreement
    - can be achieved relatively rapidly
    - die quickly



Agreements - parties

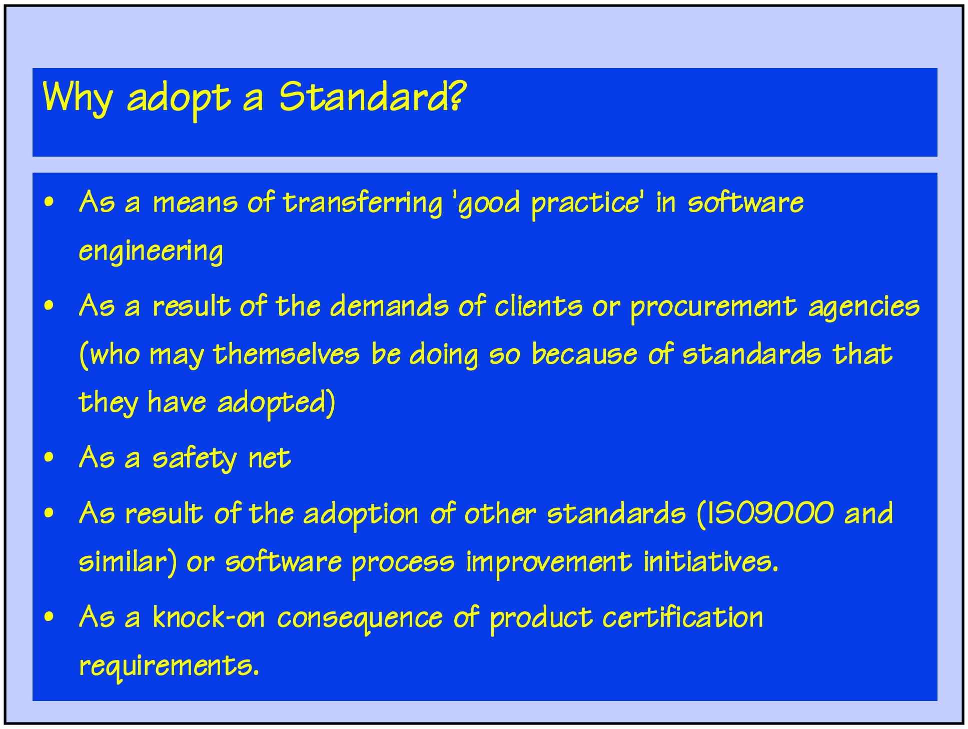
- Intra organisational
- Inter organisational
  - commercial consortia (e.g. OMG, OpenGroup)
  - professional bodies (e.g. IEEE)
- Procurer-lead
  - government (e.g. DoD)
  - large purchaser (e.g. NASA, ESA)
- Standards bodies
  - national (e.g. ANSI, DIN)
  - international (e.g. ISO, ITU)
- Open network
  - 'internet style"



Agreements - nature

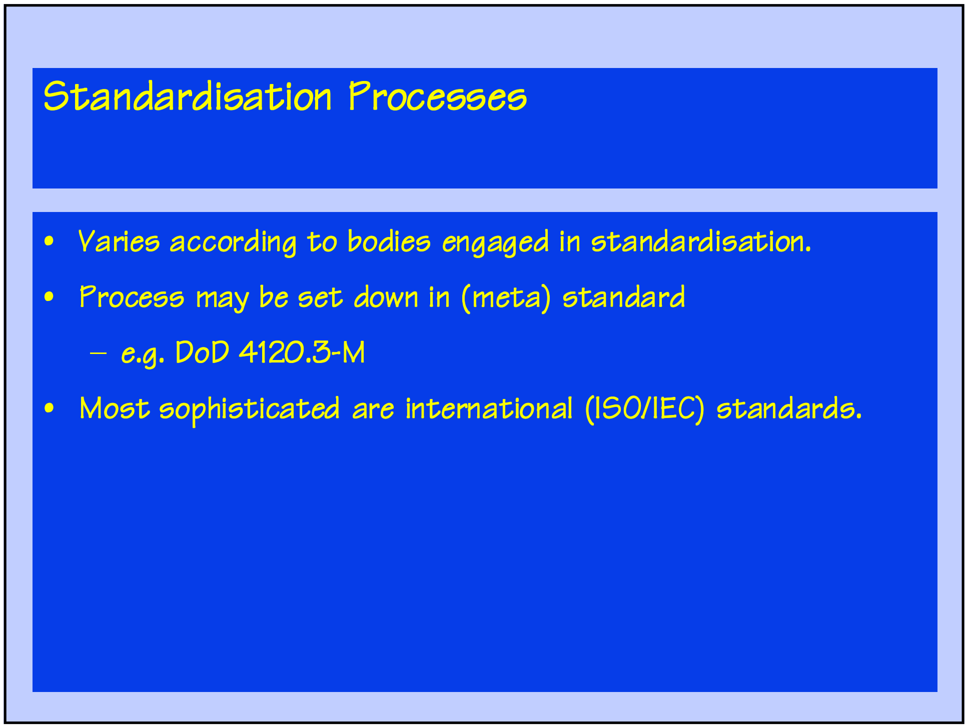
- Voluntary
  - and consensus-based
- Standards reflect maturation process of software engineering as a formal discipline.

from an art to a craft?



Why adopt a Standard?

- As a means of transferring 'good practice' in software engineering
- As a result of the demands of clients or procurement agencies (who may themselves be doing so because of standards that they have adopted)
- As a safety net
- As result of the adoption of other standards (ISO9000 and similar) or software process improvement initiatives.
- As a knock-on consequence of product certification requirements.



Standardisation Processes

- Varies according to bodies engaged in standardisation.
- Process may be set down in (meta) standard
  - e.g. DoD 4120.3-M
- Most sophisticated are international (ISO/IEC) standards.

Structure

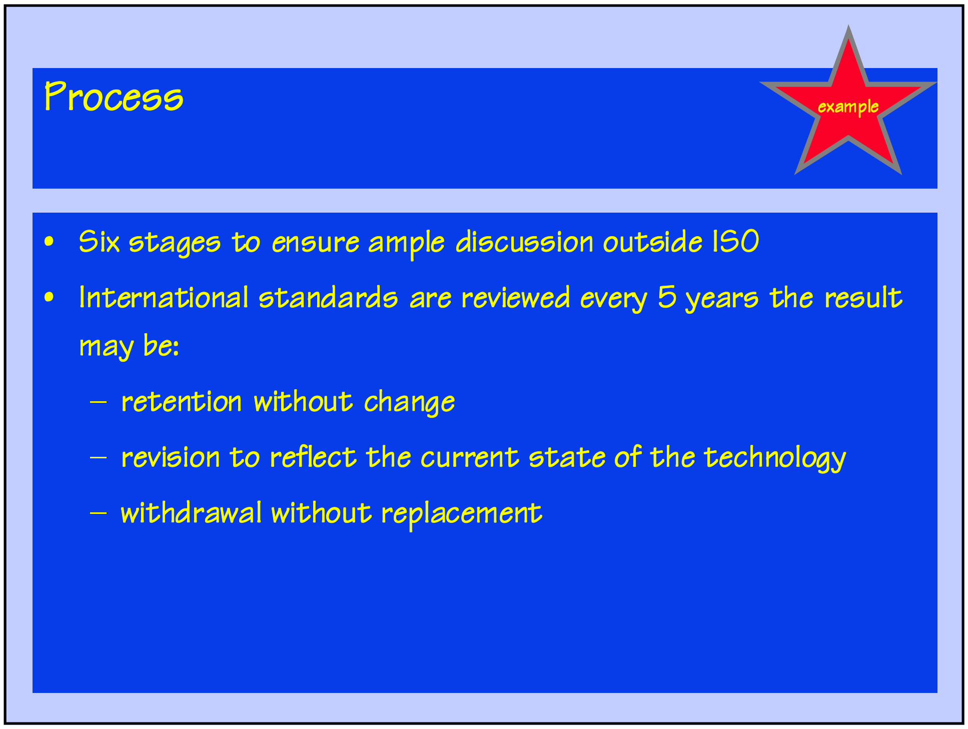
example

- International Organisation for Standardisation (ISO) and International Electrotechnical Commission (IEC) develop and promulgate standards worldwide.
- To cover IT they have formed a Joint Technical Committee (JCT1).
- JCT1 is divided into subcommittees (SC) and working groups (WG).
- Each WG is charged with the development of standards in a specialised area (there are currently 12 WGs in software engineering).

Documents

example

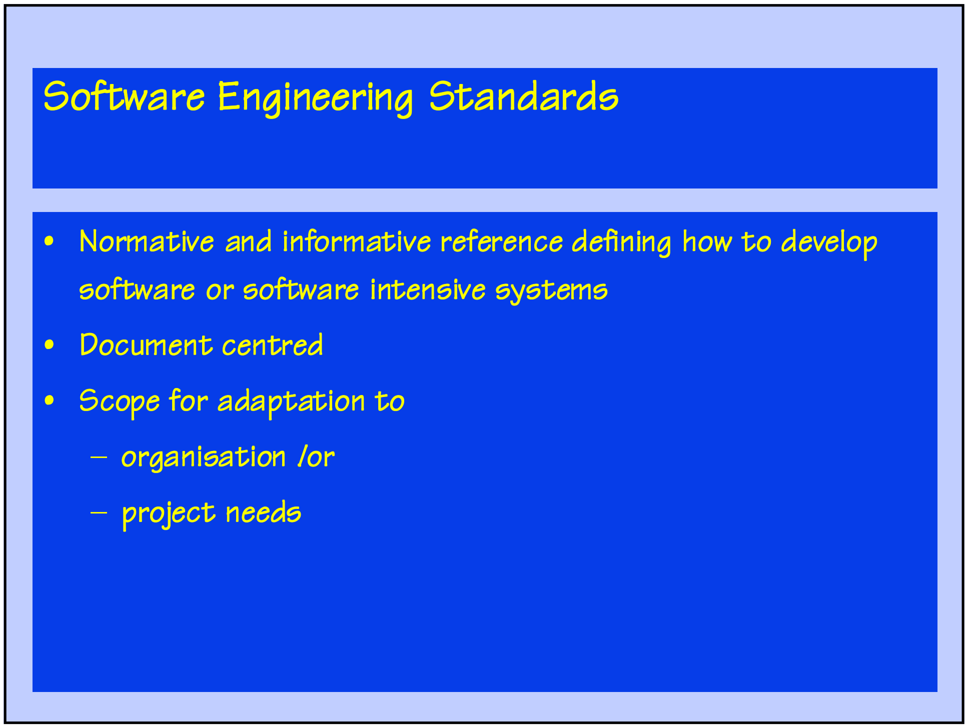
- ISO produce two main types of end documents the international standard (IS) and the technical reports (TR)
- “The social and economic long-term benefits of an IS should justify the total cost of preparing, adopting and maintaining the standard”.
- it must be demonstrated that the proposed standard is technically feasible, timely and unlikely either to be made obsolete quickly or to inhibit the benefits of technology to the users



Process

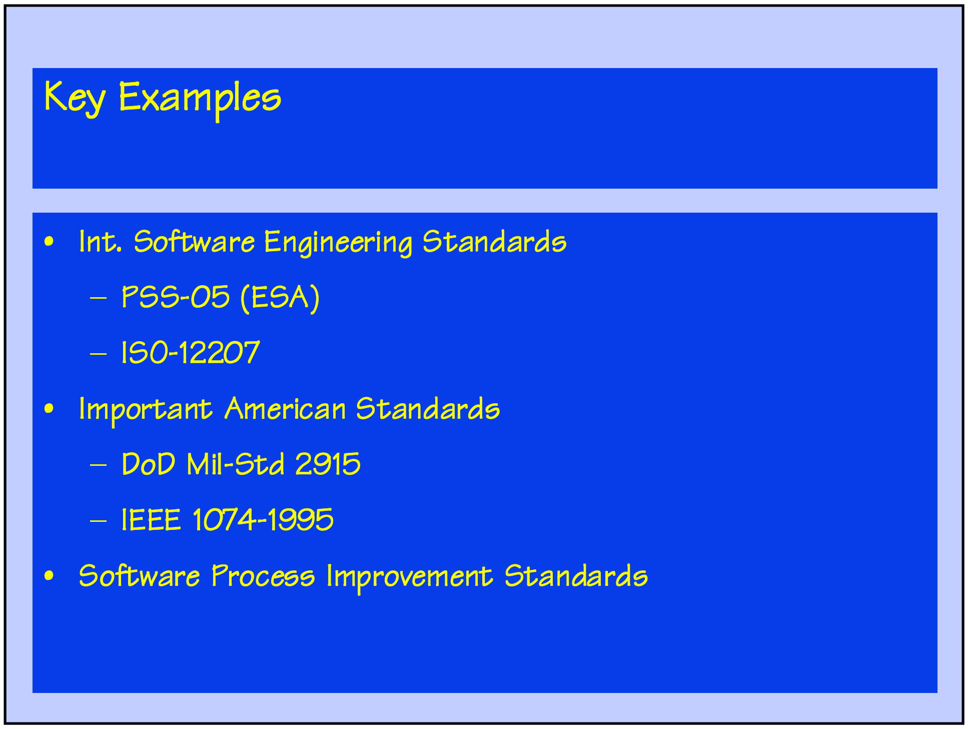
example

- Six stages to ensure ample discussion outside ISO
- International standards are reviewed every 5 years the result may be:
  - retention without change
  - revision to reflect the current state of the technology
  - withdrawal without replacement



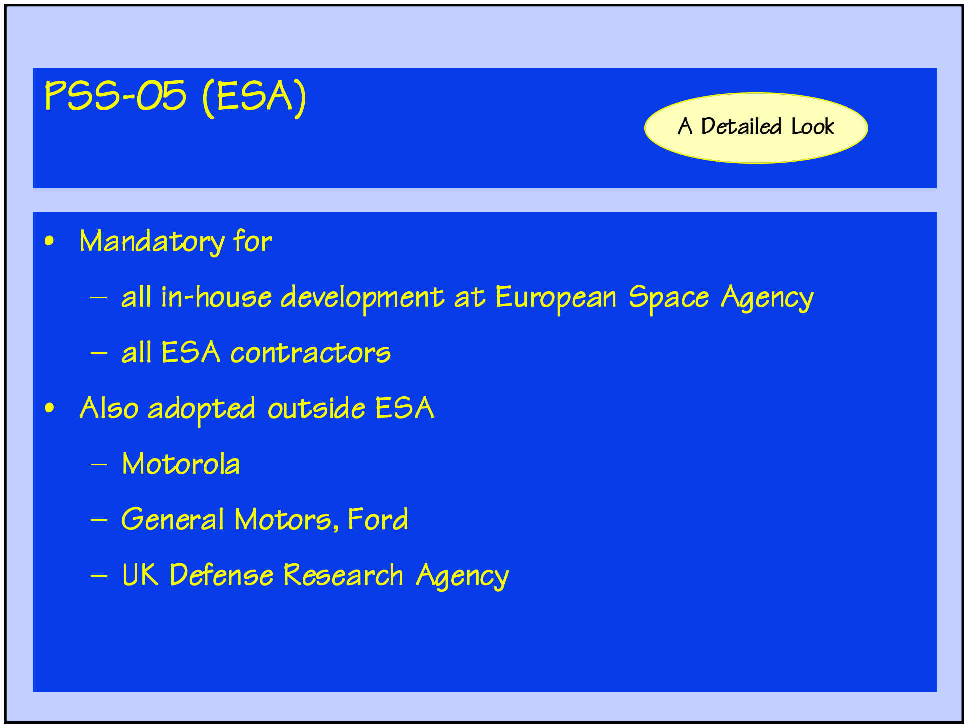
Software Engineering Standards

- Normative and informative reference defining how to develop software or software intensive systems
- Document centred
- Scope for adaptation to
  - organisation /or
  - project needs



Key Examples

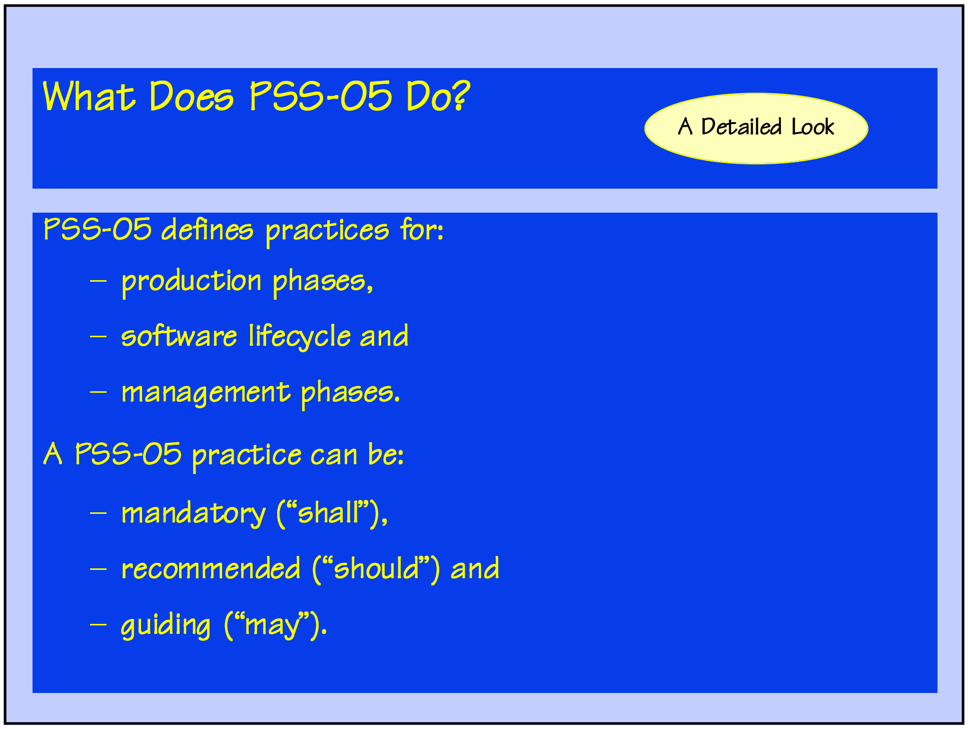
- Int. Software Engineering Standards
  - PSS-05 (ESA)
  - ISO-12207
- Important American Standards
  - DoD Mil-Std 2915
  - IEEE 1074-1995
- Software Process Improvement Standards



PSS-05 (ESA)

A Detailed Look

- Mandatory for
  - all in-house development at European Space Agency
  - all ESA contractors
- Also adopted outside ESA
  - Motorola
  - General Motors, Ford
  - UK Defense Research Agency



What Does PSS-05 Do?

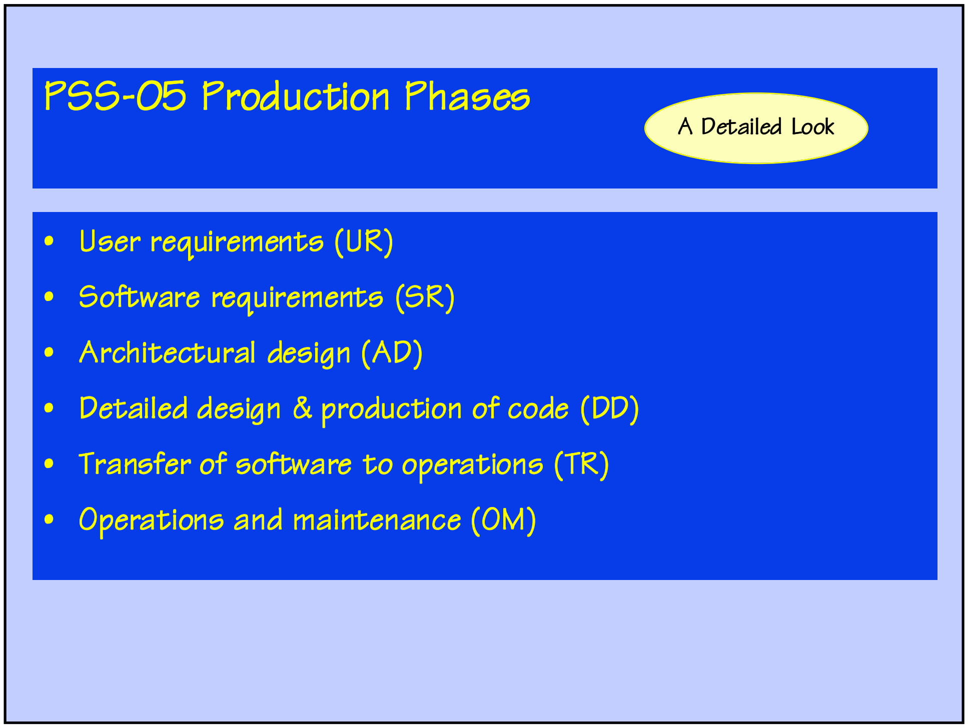
A Detailed Look

PSS-05 defines practices for:

- production phases,
- software lifecycle and
- management phases.

A PSS-05 practice can be:

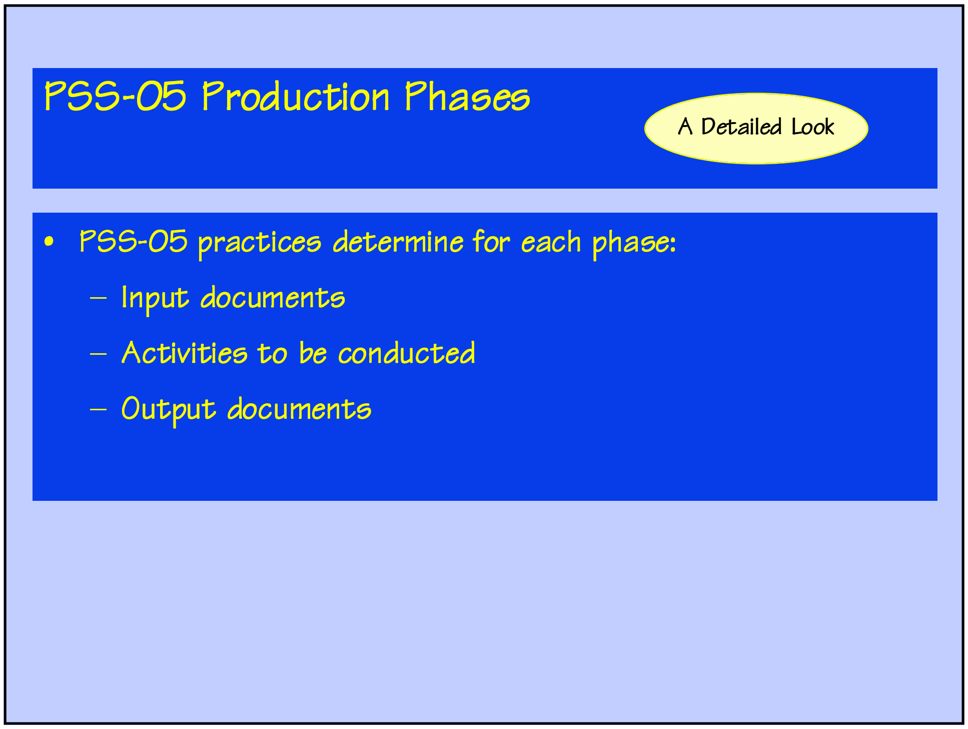
- mandatory (“shall”),
- recommended (“should”) and
- guiding (“may”).



PSS-05 Production Phases

A Detailed Look

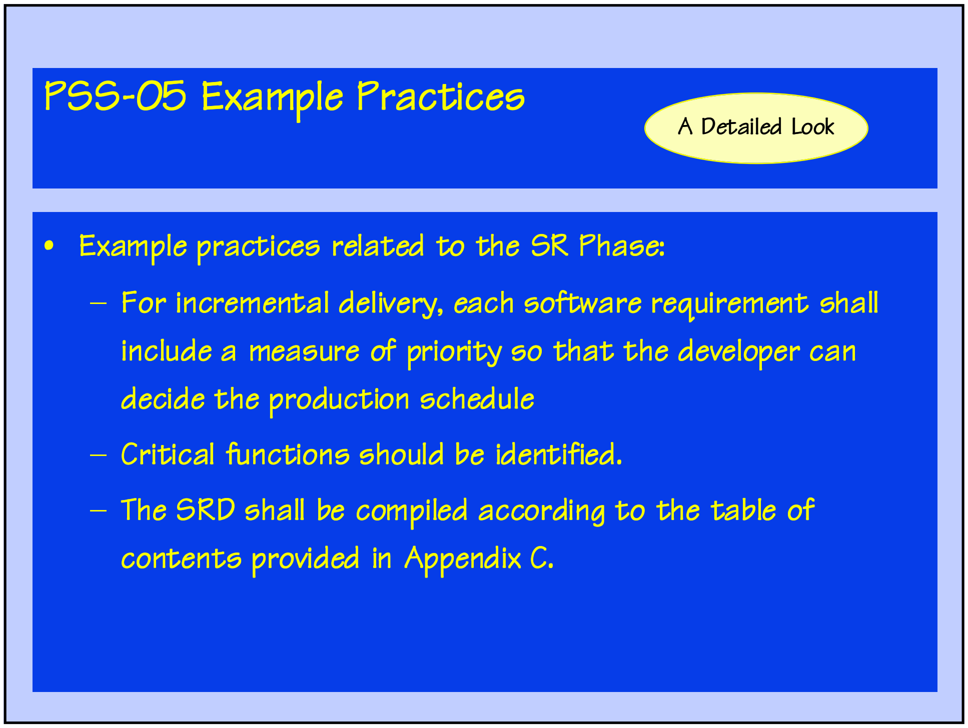
- User requirements (UR)
- Software requirements (SR)
- Architectural design (AD)
- Detailed design & production of code (DD)
- Transfer of software to operations (TR)
- Operations and maintenance (OM)



PSS-05 Production Phases

A Detailed Look

- PSS-05 practices determine for each phase:
  - Input documents
  - Activities to be conducted
  - Output documents



PSS-05 Example Practices

A Detailed Look

- Example practices related to the SR Phase:
  - For incremental delivery, each software requirement shall include a measure of priority so that the developer can decide the production schedule
  - Critical functions should be identified.
  - The SRD shall be compiled according to the table of contents provided in Appendix C.

PSS-05 Software Lifecycle

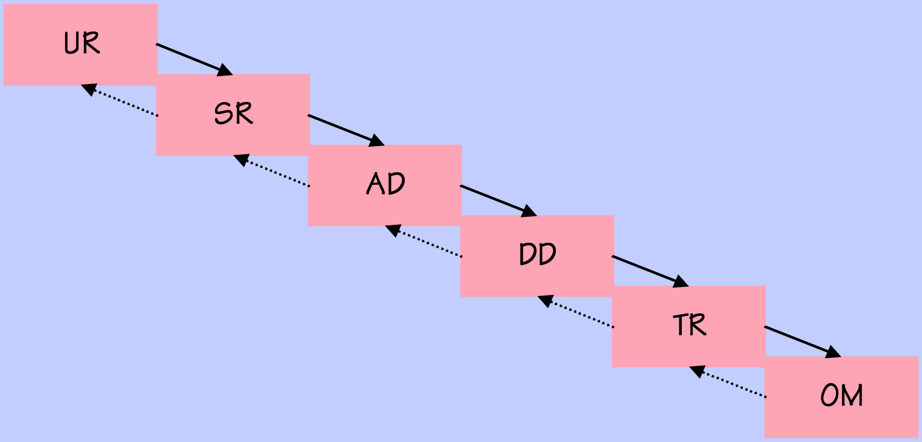
A Detailed Look

- Three lifecycle approaches are prescribed
- Process Engineer* can select one of
  - Waterfall
  - Incremental Delivery
  - Evolutionary Development

* the person responsible for "engineering" the development process

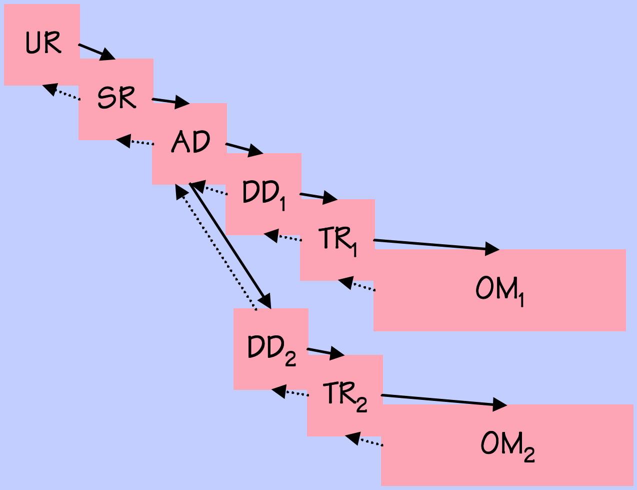
PSS-05 Waterfall Approach

A Detailed Look



PSS-05 Incremental Delivery

A Detailed Look



UR

SR

AD

DD$_1$

TR$_1$

OM$_1$

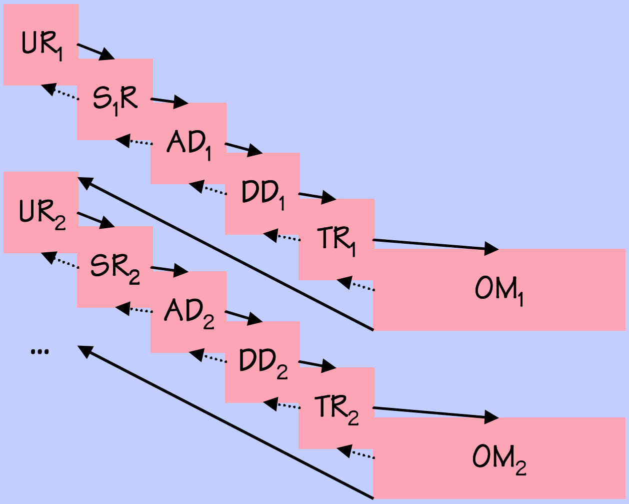
DD$_2$

TR$_2$

OM$_2$

PSS-05 Evolutionary Development

A Detailed Look



UR$_1$

S$_1$R

AD$_1$

DD$_1$

TR$_1$

OM$_1$

UR$_2$

SR$_2$

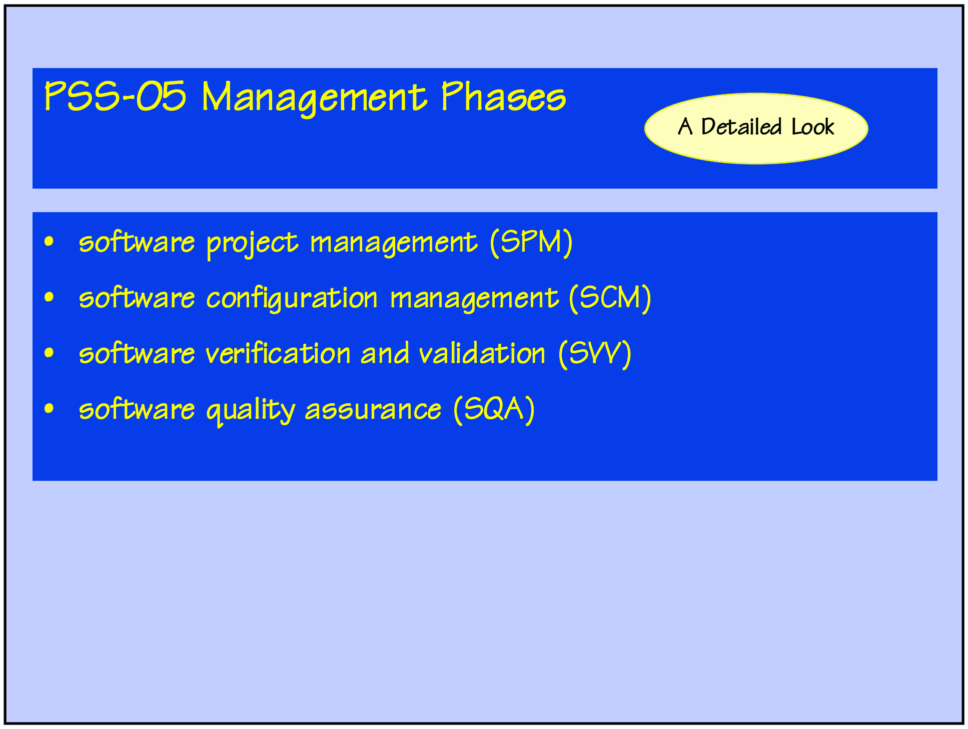
AD$_2$

DD$_2$

TR$_2$

OM$_2$

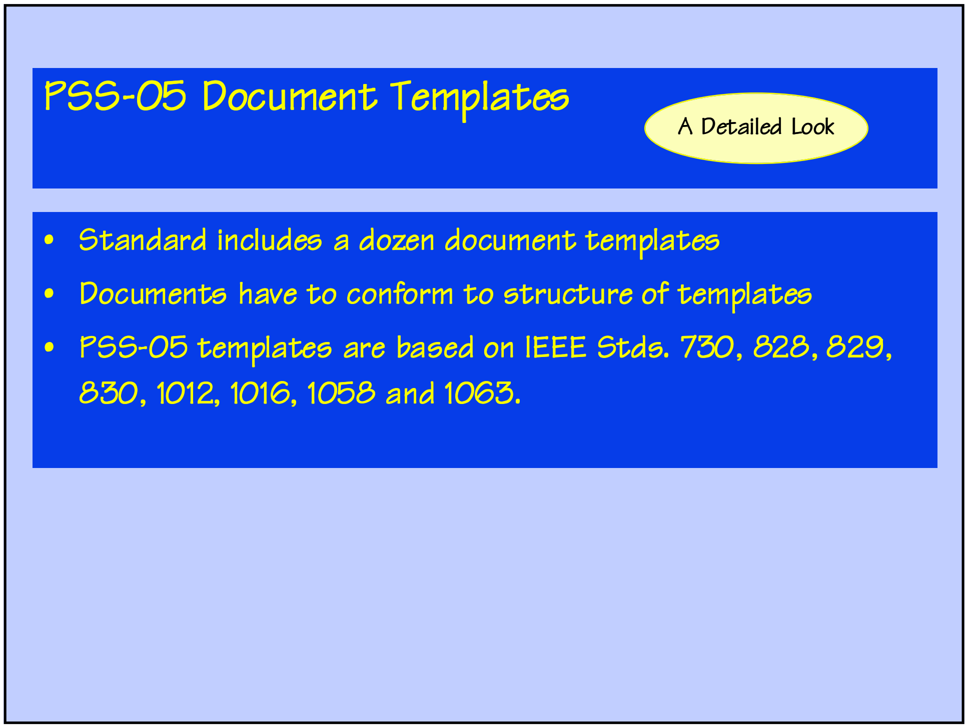
...



PSS-05 Management Phases

A Detailed Look

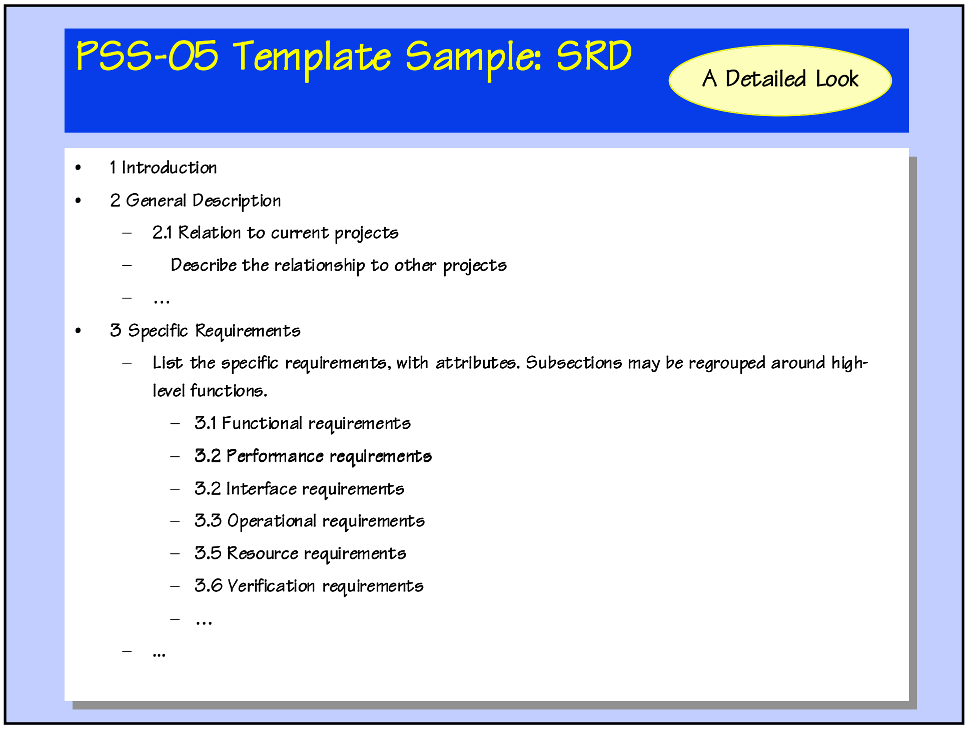
- software project management (SPM)
- software configuration management (SCM)
- software verification and validation (SVV)
- software quality assurance (SQA)



PSS-05 Document Templates

A Detailed Look

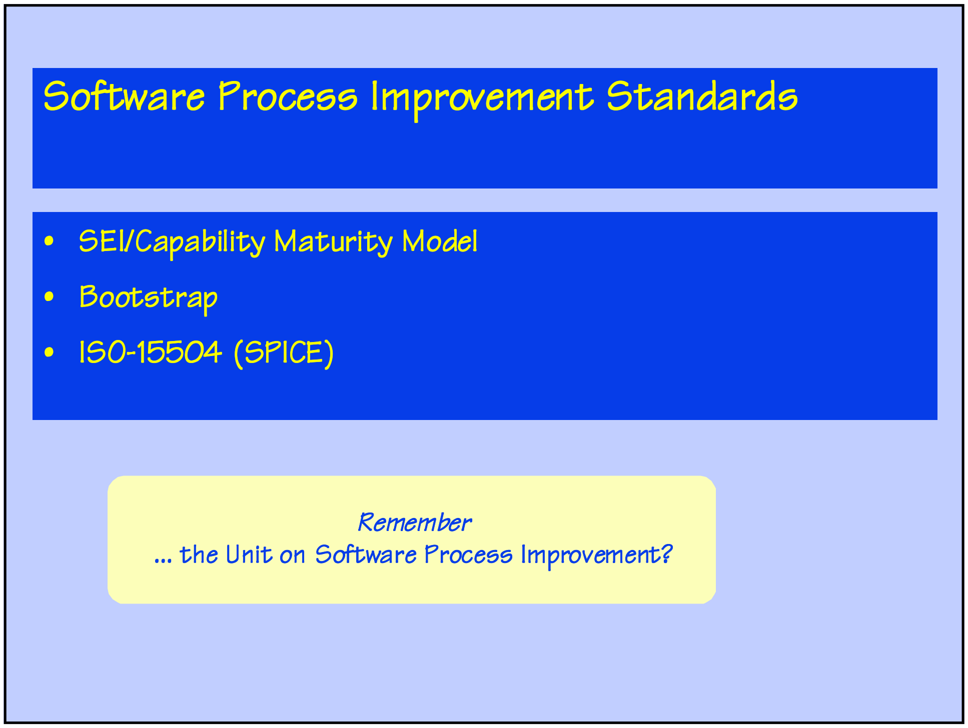
- Standard includes a dozen document templates
- Documents have to conform to structure of templates
- PSS-05 templates are based on IEEE Stds. 730, 828, 829, 830, 1012, 1016, 1058 and 1063.



PSS-05 Template Sample: SRD

A Detailed Look

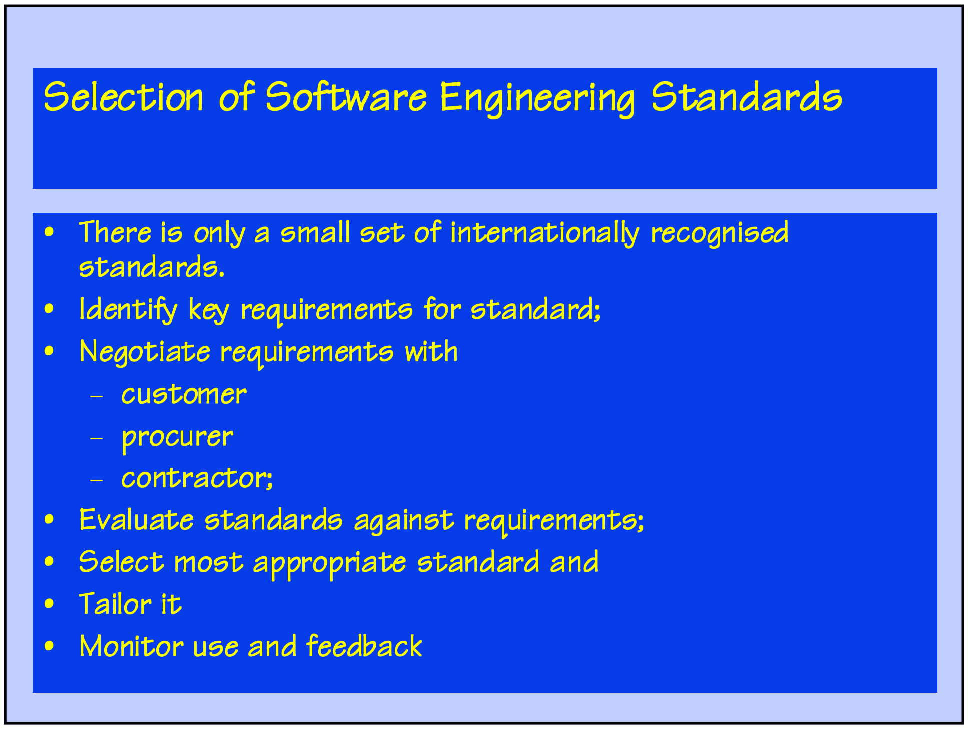
- 1 Introduction
- 2 General Description
  - 2.1 Relation to current projects
  - Describe the relationship to other projects
  - …
- 3 Specific Requirements
  - List the specific requirements, with attributes. Subsections may be regrouped around high-level functions.
    - 3.1 Functional requirements
    - 3.2 Performance requirements
    - 3.2 Interface requirements
    - 3.3 Operational requirements
    - 3.5 Resource requirements
    - 3.6 Verification requirements
    - …
  - ...



Software Process Improvement Standards

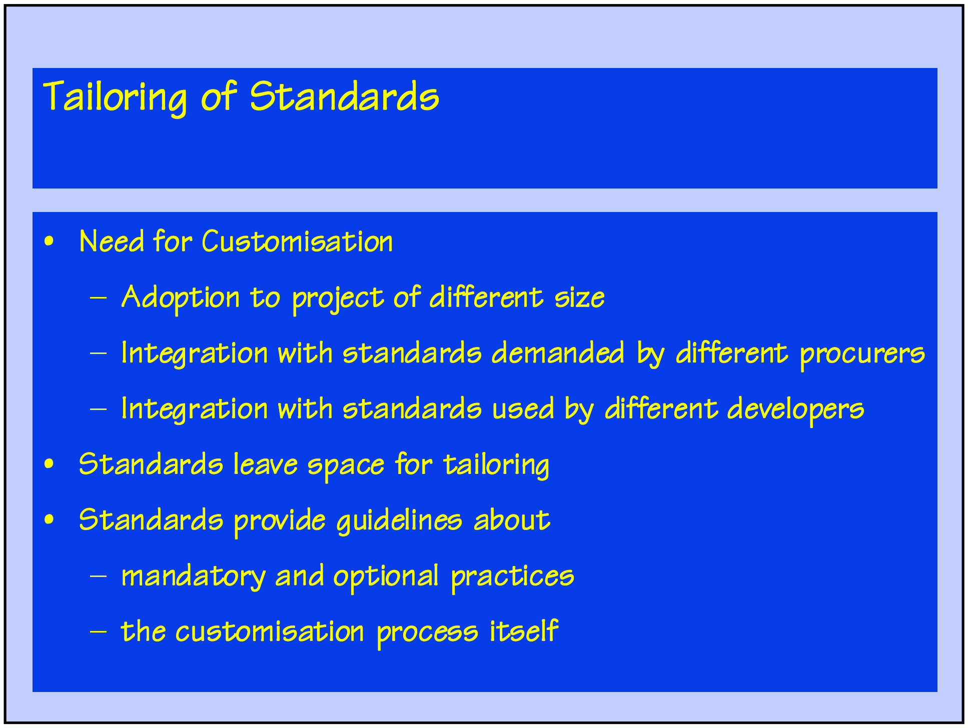
- SEI/Capability Maturity Model
- Bootstrap
- ISO-15504 (SPICE)

Remember
... the Unit on Software Process Improvement?



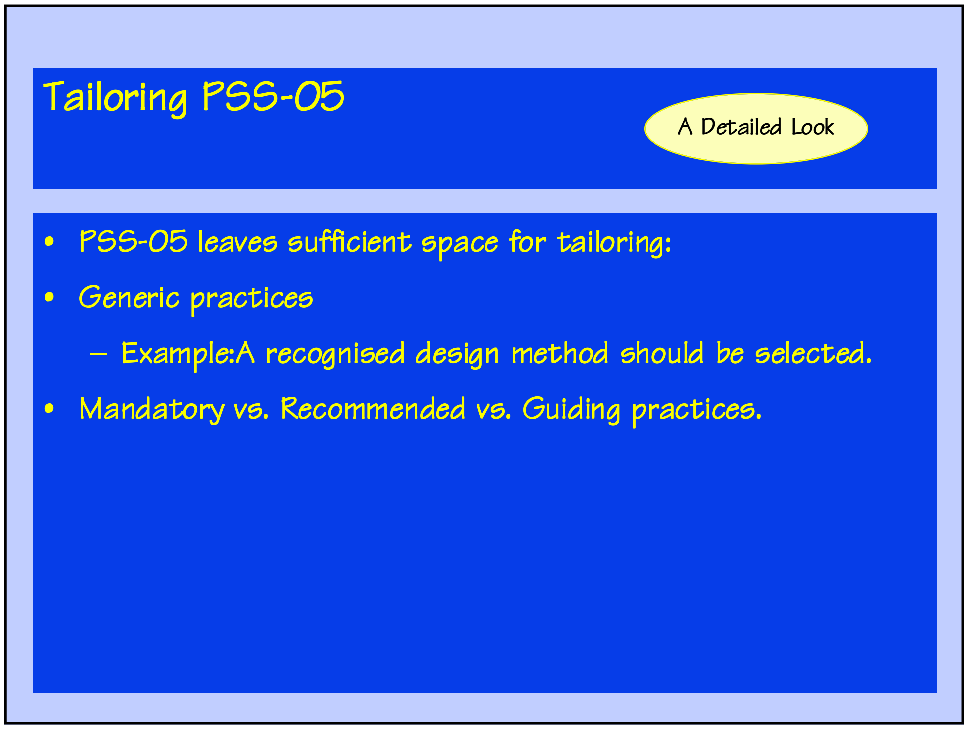
Selection of Software Engineering Standards

- There is only a small set of internationally recognised standards.
- Identify key requirements for standard;
- Negotiate requirements with
  - customer
  - procurer
  - contractor;
- Evaluate standards against requirements;
- Select most appropriate standard and
- Tailor it
- Monitor use and feedback



Tailoring of Standards

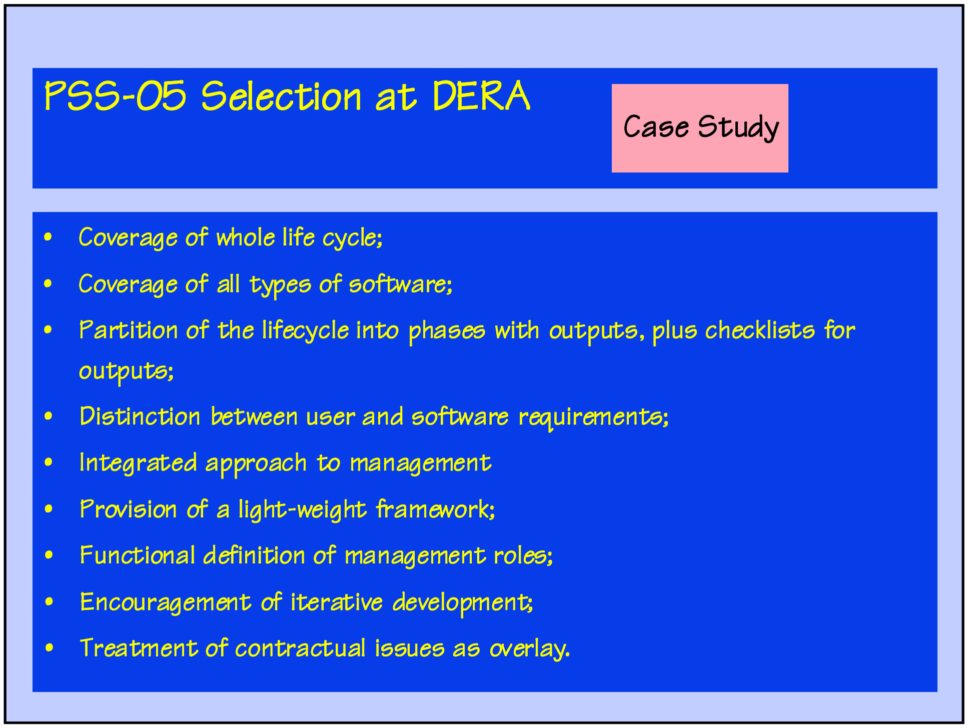
- Need for Customisation
  - Adoption to project of different size
  - Integration with standards demanded by different procurers
  - Integration with standards used by different developers
- Standards leave space for tailoring
- Standards provide guidelines about
  - mandatory and optional practices
  - the customisation process itself



Tailoring PSS-05

A Detailed Look

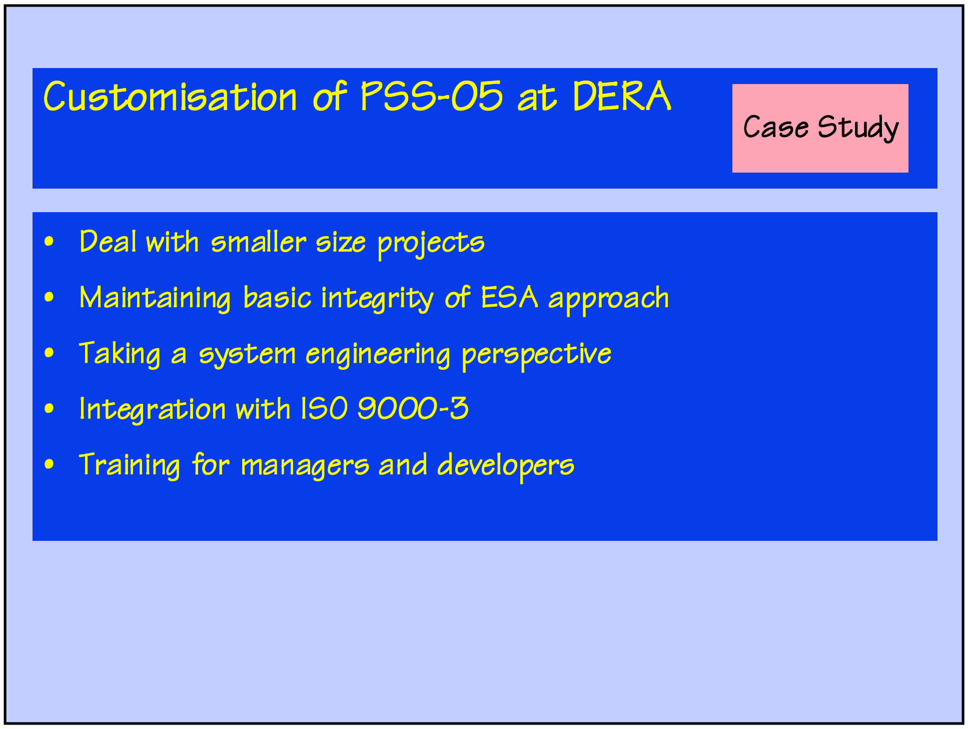
- PSS-05 leaves sufficient space for tailoring:
- Generic practices
  - Example:A recognised design method should be selected.
- Mandatory vs. Recommended vs. Guiding practices.



PSS-05 Selection at DERA

Case Study

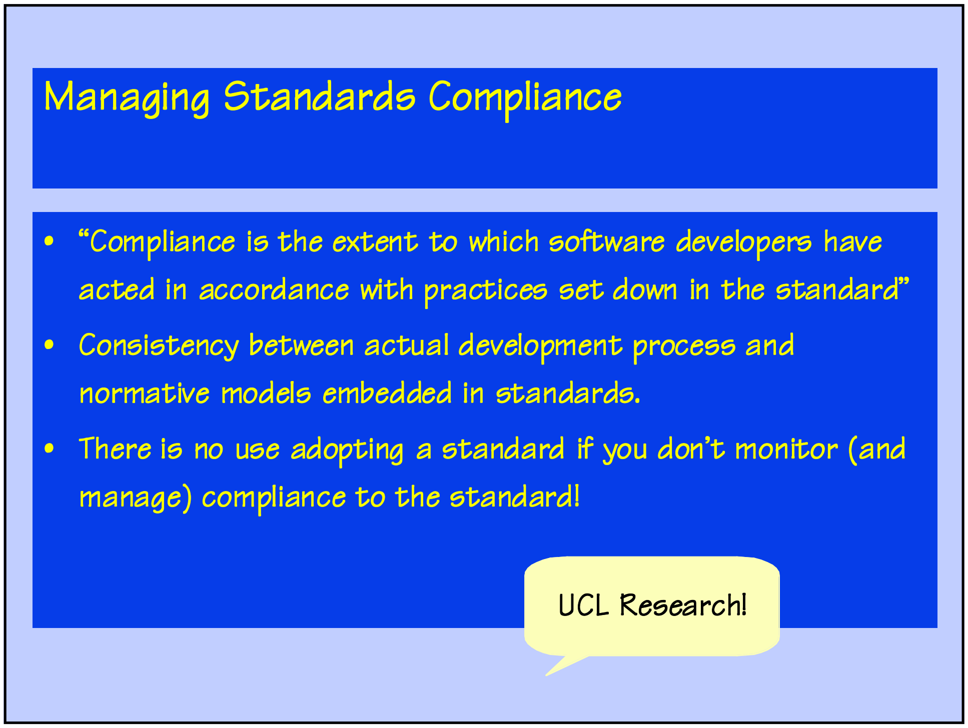
- Coverage of whole life cycle;
- Coverage of all types of software;
- Partition of the lifecycle into phases with outputs, plus checklists for outputs;
- Distinction between user and software requirements;
- Integrated approach to management
- Provision of a light-weight framework;
- Functional definition of management roles;
- Encouragement of iterative development;
- Treatment of contractual issues as overlay.



Customisation of PSS-05 at DERA

Case Study

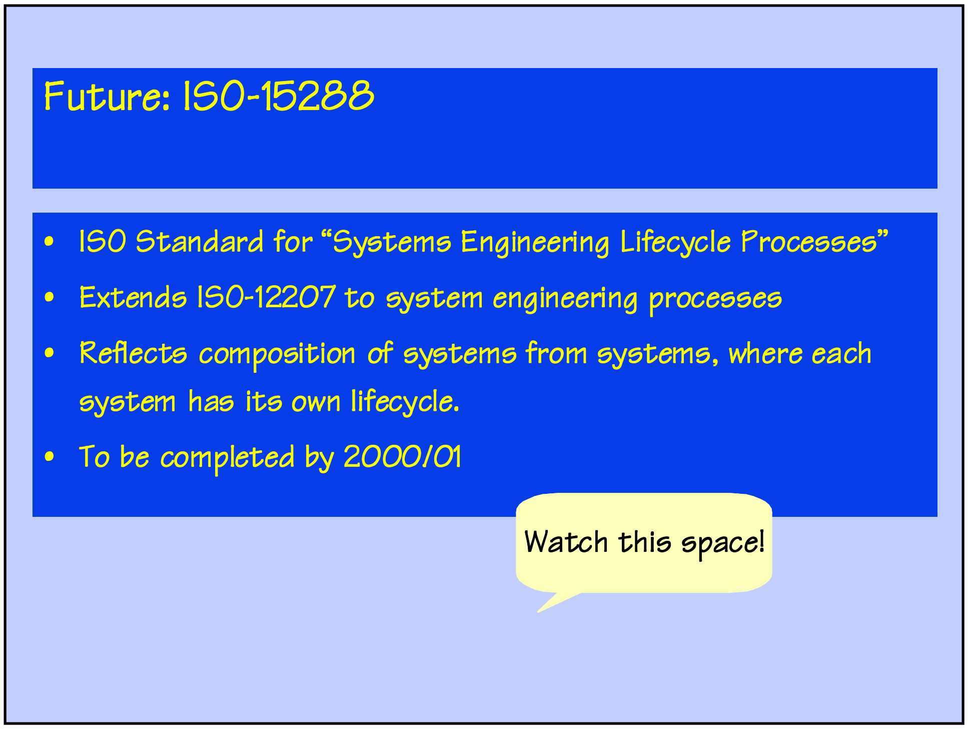
- Deal with smaller size projects
- Maintaining basic integrity of ESA approach
- Taking a system engineering perspective
- Integration with ISO 9000-3
- Training for managers and developers



Managing Standards Compliance

- “Compliance is the extent to which software developers have acted in accordance with practices set down in the standard”
- Consistency between actual development process and normative models embedded in standards.
- There is no use adopting a standard if you don’t monitor (and manage) compliance to the standard!

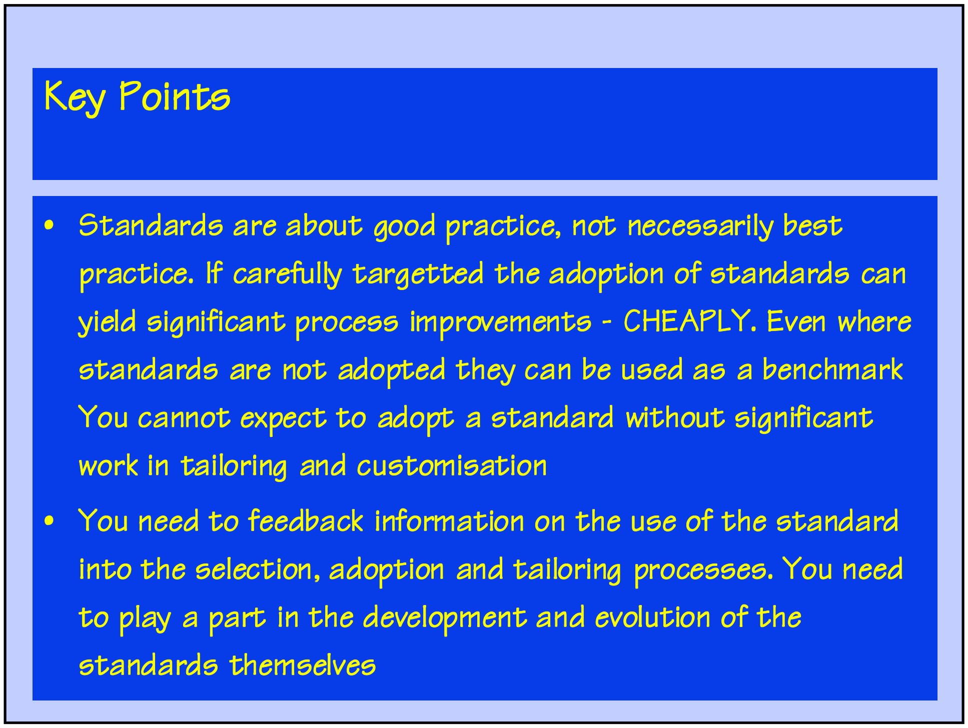
UCL Research!



Future: ISO-15288

- ISO Standard for "Systems Engineering Lifecycle Processes"
- Extends ISO-12207 to system engineering processes
- Reflects composition of systems from systems, where each system has its own lifecycle.
- To be completed by 2000/01

Watch this space!



Key Points

- Standards are about good practice, not necessarily best practice. If carefully targetted the adoption of standards can yield significant process improvements - CHEAPLY. Even where standards are not adopted they can be used as a benchmark You cannot expect to adopt a standard without significant work in tailoring and customisation
- You need to feedback information on the use of the standard into the selection, adoption and tailoring processes. You need to play a part in the development and evolution of the standards themselves