

CC5001

CC3002

**Support**

Enhancement requests

# Support issues

- **What do we need from system support?**
- **IS support service: the Help Desk**
- **Service Level Agreements**
- **Enhancement Requests**

# Enhancement requests

- Enhancement requests
  - What are they?
  - How do they arise?
  - How are they managed?
- Help Desk's role in managing enhancements
- Information Systems evolution
  - Why do IS evolve?

# What are enhancement requests?

- An event that demonstrates the need for IS change, should generate an

***enhancement request***

- Also known as
  - change requests
  - modification requests

# What are enhancement requests?

“***Change requests***... will arise from

- business managers
- users
- designers
- programmers...”

Chaffey (2003)

# What are enhancement requests?

“These requests could include

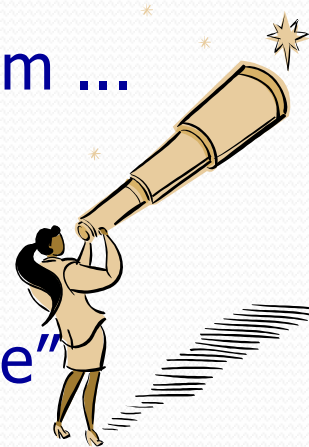
– reports of **bugs** ...



– features that are **missing** from the system ...



– ideas for **future versions** of the software”



# How do enhancement requests arise?

- Categorise as **major** or **minor** changes
- **major** changes triggered by:
  - IS (strategic) planning
  - Requests from user (business managers)
- **minor** changes triggered by:
  - Requests from user (staff)
  - Issues reported to Help Desk

# How do enhancement requests arise?

- User reports issue to Help Desk
  - user simply reporting problem
  - not aware that enhancement request triggered
- Help Desk identifies need for enhancement





# How do enhancement requests arise?

- Many issues resolved ***without*** changing IS
  - ***some*** issues ***will*** require change to software
- Enhancement request is generated
  - defined procedure then followed

# Managing enhancement requests

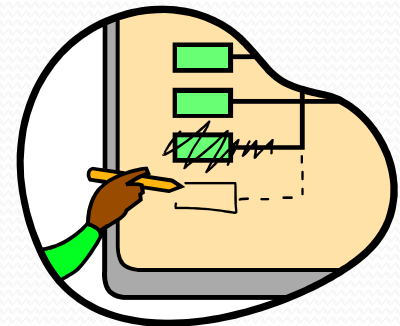
- **Major changes**
  - trigger the initiation of a ***new project***
- Project life cycle starts again...
  - conceptualisation
  - planning
  - implementation
  - completion



# Managing enhancement requests

- **Minor changes**

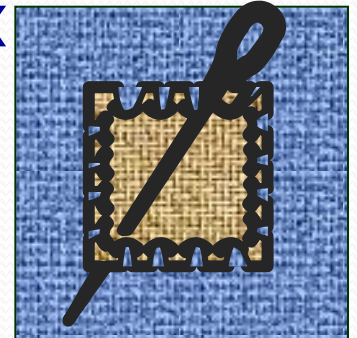
- Initiation of a small / “mini” project
- Focus on area requiring change



# Managing enhancement requests

- **Minor changes**

- Immediate action to supply a fix
  - known as *a patch*



**Software patch:**

*“... an interim release of part of an information system that is intended to address deficiencies in a previous release”*

(Chaffey, 2003)

# Managing enhancement requests

- **Minor changes**

- Change is bundled in with other minor changes

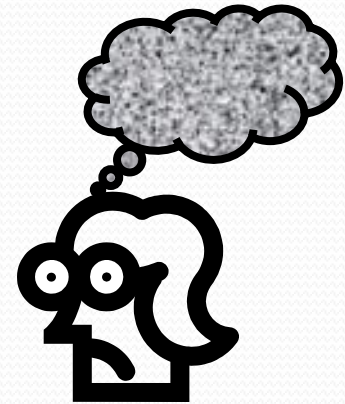
- delivered in next ***maintenance release*** of software



# Managing enhancement requests

- **Minor changes**

- Change request is rejected



# Software releases, versions, etc.

- Software releases indicated by version no.
- **Major** release
  - will include many new features, etc.
  - version number changes
    - e.g. 1.1 to 2.0



# Software releases, versions, etc.

- Software releases indicated by version no.

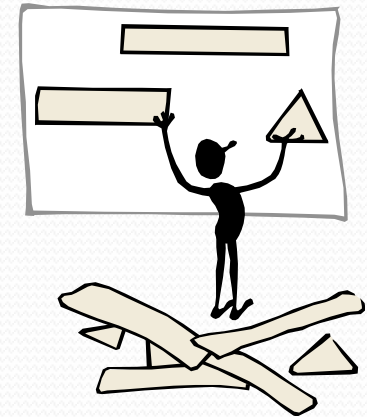
- **Minor** release

- may include

- some new features
    - fixes a number of problems
    - release of bundled change requests, etc.

- version number changes

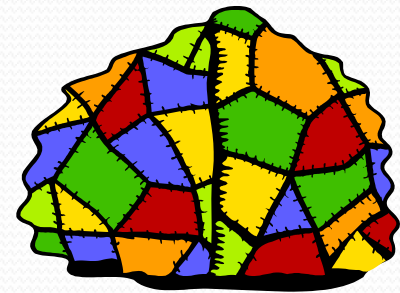
- e.g. 2.0 to 2.1





# Software releases, versions, etc.

- Software releases indicated by version no.
- **Patch** or interim release
  - fixes one or more problems
  - version number changes
    - e.g. 2.1 to 2.11

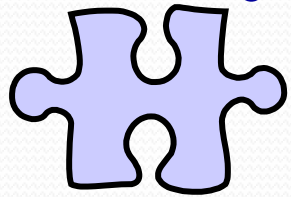


# Configuration management and change control

- Configuration management
  - the control of the *different versions* of the various software components (or products) comprising an IS
  - products may include hardware, O/S, software, DB, documentation, etc.

from Cadle & Yeates (2001)

# Configuration management and change control

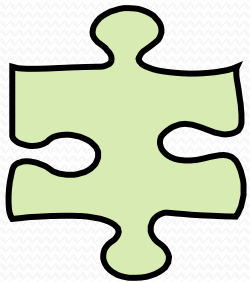


- Configuration management

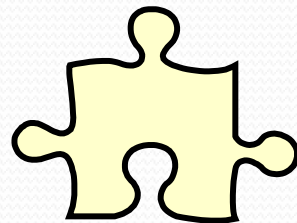
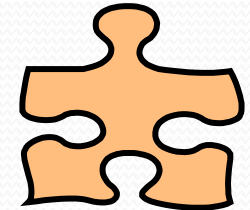
- BS8488 (1984) definition

- “the discipline of identifying the components of a *continuously evolving system*

- (taking into account relevant system interfaces)



- for the purposes of *controlling changes* to these components and *maintaining integrity* and traceability...”



# Configuration management and change control

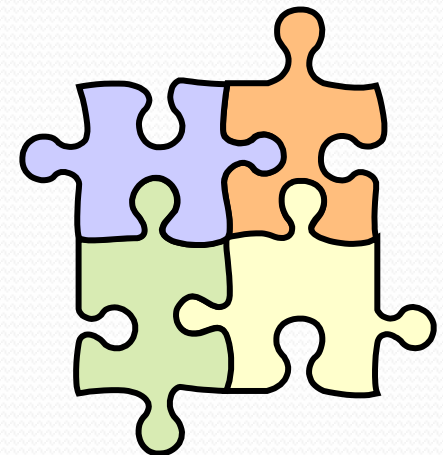
- Configuration management involves
  - **identifying** changes to products
  - **controlling** changes to products
  - ensuring changes are made **properly**
  - **reporting** changes to others

(Beynon-Davies, 2002)

**Change control:** managing changes to an IS, including impact analysis, authorisation and documentation

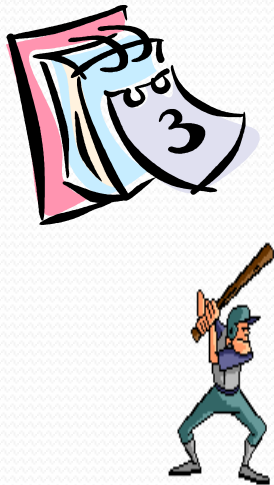
# Configuration management and change control

- Configuration management involves
  - understanding **dependencies** between different components
  - **compatibility** of different versions of componentsof an Information System



# Impact analysis and regression testing

- Before any change to an IS is authorised
  - analysis of the impact of the change made



- **cost** of the required development work
- **time** for the required development work
- effect of the change on (all) the **users**
- effect of the change on other parts of the **IS**
- consequences of **not** implementing the change



# Impact analysis and regression testing

- **After** development work completed



- **appropriate regression testing**

- **Before** a change is installed



# Regression Testing

- Testing program changes
  - ensure that ***old code still works*** with the new changes
- May be done by code testing specialists





# Regression Testing

- Coders develop code test scenarios and exercises
  - test ***new units*** of code written
  - test cases form ***test bucket***



Regression testing 2007

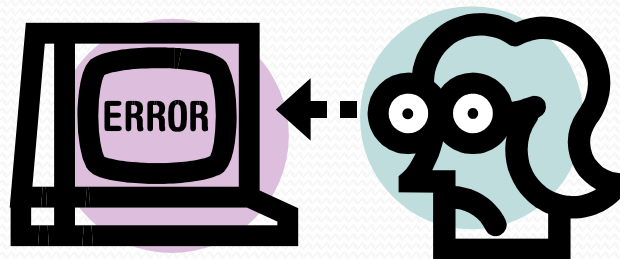
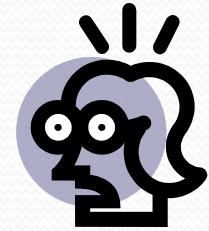
# Regression Testing

- **Before** a new version of software is released
  - test bucket cases run against the **new version**
  - make sure that all **old capabilities** still function correctly



# Regression Testing

- *What could go wrong...?*
- System might not work
  - changing code
  - adding new code
- introduce ***new errors*** into code
- make ***unintended changes*** elsewhere



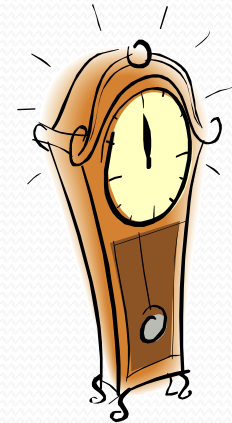
Regression testing 2007

# Information Systems evolution

- New systems do not remain static

– ***IS change over time***

- IS “evolve” in response to environmental changes



# Information Systems evolution

- Reasons for evolution

## **Problem with IS**

- Poorly developed initial system

- requirements not met
- technical problems (bugs) in system

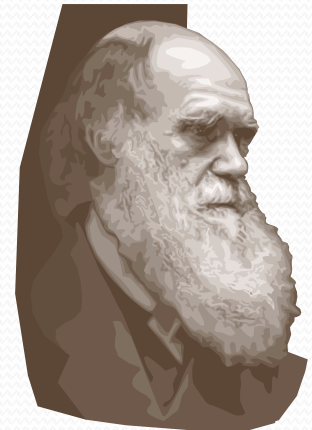


# Information Systems evolution

- Reasons for evolution

## **Change in the environment of the IS**

- Business process changes
- Changes in the environment
  - legislation
  - software
- Technical environment changes
  - policy decision to move to client-server architecture
- Changes in related IS



# Information Systems evolution

- Reasons for evolution

## **Success of IS**

- A successful IS stimulates further change
  - generation of ideas for new systems
  - extending systems to new contexts



# Summary

- Enhancement requests
  - Major / minor changes, patches, versions
- Configuration management
- Change control
- Impact analysis
- Regression testing
- IS evolution







**Any questions?**

# References and further reading

- Beynon-Davies, P. 2002, *Information systems*, Palgrave
- Chaffey, D. (ed.) 2003, *Business Information Systems*, 2nd edition, FT Prentice Hall
- Jorgenson, P. 1995, *Software Testing: a Craftsman's Approach*, CRC Press - cited in Chaffey (2003)
- Czegel, B. 1999, *Help Desk Practitioner's Handbook*, Wiley
- Help Desk World, 2002, *Help Desk Software World:What is a Help Desk?* Retrieved: 11 March 2013 from <http://www.help-desk-world.com/help-desk.htm>
- Microsoft, 1997, *Microsoft Sourcebook for the Help Desk*, 2nd edition, Microsoft Press International
- Mohr, J (nd), *The Help Desk*. Retrieved: 11 March 2013 from [http://www.jimmo.com/modules.php?name=Content&pa=list\\_pages\\_categories&cid=11](http://www.jimmo.com/modules.php?name=Content&pa=list_pages_categories&cid=11)
- Tourniaire, F. & Farrell, R. 1997, *The Art of Software Support*, Prentice Hall
- Regression Testing 2007 Retrieved: March 11, 2013, from <http://searchsoftwarequality.techtarget.com/definition/regression-testing>