UWA Centre for Software Practice

Dr David Glance
Background: Me

- 20 or so years in industry – finance and software companies – Tibco, Reuters, Iona Technologies and Microsoft
- PhD in Physiology – computing a secondary interest
- Research interests: eHealth, eResearch, Bibliometrics
Background: CSP

- Started in 2005 as Rapunzel Web
- Made a University Research Centre in 2006
- Design and Development
  - Websites
  - Software for Tertiary Sector
  - Software for eHealth
- Research & Research Training
  - 1 Postdoc and 1 PhD student
  - eHealth
Staff & Students

- Staff
  - 9 Full time staff
  - 2 part time
  - 1 postdoc
- 7 former students/staff now working in industry
- About 12 students have done practicums with CSP
Development Environment

• C# ASP.NET Ajax SQL Server
• Mostly web-based applications
• V-Agile Process Methodology
• Bespoke software (at UWA but increasingly external)
• Commercial software (Socrates and MMEx)
• Largest project – managed health network in WA for $4.0 m – now contract to the Department of Health WA
CSP Typical Development Process 1

- Requirements are captured in iterative process involving wireframes
- Wireframes work well but users never really know what they want until they start using a product
- Scope of the project is roughly agreed when price is discussed – this becomes the limiting factor to functionality of the system
- Important to remember that most software continues development – everything is a milestone along the way
CSP Typical Development Process 2

- Bug and issue tracking done using issue tracking software (was gemini now done through software eSupport)
- Encourage end-users to enter bugs/issues and close them off once they have been fixed
- Also “instrument” code so that exceptions (errors) are automatically emailed to the developer
  - Errors can be fixed before they are reported
  - Errors should be entered into bug tracking but if fixed as a result of notification, not always
  - Absence of errors not necessarily an indication that the code is error-free
CSP Typical Development Process 3

- Source code control used as standard (was Visual Source Safe now subversion)
- No specific build process (Build incorporated into Visual Studio)
- No systematic testing (no testing frameworks)
- Users are warned of the consequences of releasing software early and often – not normally a problem if issues are resolved quickly
- Perpetual Beta
CSP Typical Development Process 4

- **Pros**
  - Rapid development with little overhead
  - Works really well if good relationship with user
  - User has to want to work this way
  - Product closer to user’s expectations

- **Cons**
  - Hard if users unreasonable
  - Little proper testing and so issues crop up
  - Only works if you can refactor all the time
  - Hard to convince people who love project management that it is an effective way to develop
  - Code is the documentation
Projects

- Neptune: SURF Online
- Socrates: Research Quality Management System
  - Integrates data from multiple sources to construct staff research profiles
  - Sold to University of Notre Dame Australia
- MMEx: Medical Message Exchange – used by GPs, Hospitals, Pharmacists, etc to send patient information between each other – being used increasingly throughout WA and rest of Australia
What’s next?

- Recruiting good programmers
- Research focus
- Development of MMEx product
- Continue to have fun