A software tool supporting a constructivist approach to assessing student team work in software development

By Helen Partou
h.partou3@herts.ac.uk
and Lindsay Smith
l.1.smith@herts.ac.uk
School of Computer Science
School of Computer Science (SCS) students - ‘Zero to Hero’ in six weeks

• Preparing students for employment in the computing industry
  • Existing software tool (old platform): not ‘fit for purpose’
    • Technical components not scalable for new programming languages
    • Overly complicated to use and difficult to adapt
  • Relevant software engineering experience(s)
    • Context of software development is core delivery
      • Software tool (new platform) piloted in 2015-16 L7 Sem. A module
        • Expanded into L5 modules including online provision in Sem. B
      • Approx. 850+ students have now experienced the platform
Platform structure & demo

• Portable web-based platform
  - Robust: accessible with few unrecoverable technical failures
• Model-View-Controller (MVC) chosen architectural structure
  - Separates database, visual elements and programming interactions
  - *Facilitates teamwork:* version control compatible, e.g. Dropbox
Assessment strategy – solving a problem

• Complexity of software development presents specific educational challenges for SCS students
  • ‘Soft’ skills focus on team working
    • Passengers (lack of interest, engagement and/or feeling of inferiority) vs. diligent isolation (poor delegation, perfectionism and/or presence of passengers)
  • ‘Hard’ skills focus on technological constraints
    • Time constraints: platform minimises technical complexity for development of solution application

• Industry value: why code in teams?
  • Software developers cannot put graduates ‘in front of a client’ [1]
  • QAA Computing benchmarks: software ‘exposure’ and ‘substantial’ group projects [2]
Assessment marking criteria – guide to team project management

• User Acceptance Tests (UATs)
  • check software is ‘fit for purpose’

• For **Assessment** (tutors simulate client) categorised marking criteria
  • **Baseline** = minimum engagement for a pass mark
  • **Advanced** = independent tasks gain higher marks

• UATs support delegation of tasks to team members
Constructivism & Instructional Scaffolding

• Platform applications scalable to virtually any ‘real-world’ scenario
  • **Formative**: ‘Orders’ system included in platform
    • Minimises ‘expectation gap’ [3]
  • **Summative**: Olympic games, resourcing school productions, smart tech, etc.
    • Students ‘construct’ ideas

• Instructional Scaffolding:
  • Practical guides, FAQs, demonstration videos and hands-on lab supervision
    • Supports VARK (Visual/Auditory/Read-Write/Kinaesthetic) learning style
Critical reflections

• Future-proofing platform delivery
  • Industry-standard technologies
    • The Cloud, e.g. Git-based tools
  • Proactive planning: staff development time and resources
    • Current platform = 500+ staff hours (conservative estimate)

• Managing student teamwork autonomy
  • Staff familiarity, e.g. level of staff involvement in student teams

• Instructional scaffolding affected by VLE constraints
  • Students can face a challenge accessing teaching resources

• Criticality of case study for platform to support constructivism
References

