
Joy Garfield
j.garfield@worc.ac.uk

Abstract
Systems Analysis and Design as a taught subject at undergraduate level can be challenging for both the lecturer and student. The large number of theoretical concepts that need to be taught and assessed and the difficulty of simulating these within the classroom environment can result in dull teaching sessions and disengaged students. This paper puts forward a scaffolded case study approach for assessing undergraduate level Systems Analysis and Design. Student module evaluation feedback and grades are presented to show its effectiveness for engaging students in the Systems Analysis and Design discipline.

Keywords
Systems Analysis and Design, assessment, case studies, undergraduate, group work, scaffolding.

Introduction
Although a relatively old field, Systems Analysis and Design remains fundamental to the computing discipline as a way of developing more effective and efficient systems (Dennis et al., 2015). Graduate job roles within the field show that salaries for a number of areas relating to Systems Analysis and Design have increased significantly over the last twelve months as reported by IT Jobs Watch (2017) emphasising the demand and need for preparing graduates to fulfil such roles (Erturk, 2014): junior systems analyst +32%; agile requirements analyst +16.27%; Structured systems analysis and design method +36.36%; software development analyst +25%; information systems analyst +14.66%; object-oriented analysis and design +15.78%. Furthermore Systems Analysis and Design tends to provide a foundation for other subjects within computing based degrees.

However Systems Analysis and Design can sometimes be described as a ‘dry’ subject due to its theoretical nature (e.g. Rob, 2006). In contrast students tend to enjoy the more ‘hands on’ practical elements of taught sessions rather than those that are theoretical (Rob, 2006). Indeed Systems Analysis and Design concepts are “difficult to practice in the classroom environment” (Chen, 2005). Consequently Chen (2005, p.1) notes that “Systems Analysis and Design teaching is notably more difficult and challenging than teaching some other fields in computer science and computer information systems”. Furthermore Guy et al (2000) notes that traditional skills in modelling data and processing of operations performed by a single user interacting with a computer are no longer enough to train effective software analysts and designers. And the use of traditional lectures, seminars and labs can struggle to provide these skills (Oh Navarro and Van der Hoek, 2005).

This paper puts forward a scaffolded case study approach for assessing theoretical concepts in Systems Analysis and Design at undergraduate level based on a case study at a UK university. The remainder of this paper is organised as follows: relevant literature within the field relating to Systems Analysis and Design learning, teaching and assessment is discussed together with the case study approach, the
scaffolded approach is outlined, module results and feedback demonstrating the effectiveness of the approach is given and conclusions drawn.

Relevant Literature

Research shows that Systems Analysis and Design is taught and assessed in a variety of ways, of which some researchers advocate certain techniques, e.g. case studies (Davey and Kelly, 2005); development of an end-product such as a web-based project (Kovacs and Rowell, 2001), problem based learning (Fatima and Abdullah, 2013); group projects to improve students soft skills (Nance, 1998); flipped classroom approach (Tanner and Scott, 2015); an agile approach (Pieters, 2013); games-based learning (Tepper, 2014); and simulations (Kerins, 2012) amongst others.

Freire (1984) argued that any pedagogy should have demonstrable relevance to the immediate worlds of the students and must enable them to analyse, theorise and intellectually engage with those worlds. Furthermore for teaching to be effective, deep learning (i.e. fully understanding concepts) needs to take place (Marton and Saljo, 1976). Biggs and Tang (2011) note that this occurs when the student is motivated to understand and engage with taught concepts and are cognitively engaged through thinking, reasoning, analysing and/or problem solving. This research uses scaffolding (Bruner, 1966) as a way of promoting a deeper level of learning and providing support to promote learning when Systems Analysis and Design concepts and skills are first introduced to students. The supports are then removed as students develop autonomous learning strategies.

The Case Study Approach

Case studies in particular can enable a deeper understanding of information systems phenomena, e.g. Walsham (1993). Their well suited characteristics enable the understanding of interactions between Information Technology related innovations and organisational contexts (Darke et al., 1998) together with the understanding of complex phenomena (Yin, 2003), the capturing of ‘reality’ in greater detail and analysing more variables than is possible by using laboratory experiments, field experiments and surveys (Galliers, 1992). Indeed there is no single way of conducting a case study and a combination of methods can be employed, such as interviews and study of secondary data.

Although case studies have been traditionally used for over 30 years by teachers in disciplines operating in a business working environment (e.g. Barnes et al., 1994), case studies used in higher education that are text based, simplified, static and not interactive or immersive can give students limited access to a real-life situation (Guy et al., 2000). However it is important that students gain knowledge of the complex social and business environments in which business systems are developed (Cope and Horan, 1996).

Project based approaches for computing courses can be productive in enhancing student learning in course concepts (e.g. Yousif and Naghedolfeizi, 2007). “In working life almost all substantial activities are undertaken as team efforts, and there will seldom be an opportunity to work alone” (Cornford and Smithson, 2006). Cornford and Smithson (2006) also highlight other beneficial aspects of group work which include the showcasing of talents of group members, use of debate and discussion within the group to refine and improve the quality of work together with building and cementing friendships. Furthermore collaborative learning can enhance critical thinking through discussion, clarification of ideas and evaluation of others’ ideas (Gokhale, 1995) and enables students to build and practice skills that are a necessary requirement for future employment (Wells and Jones, 2005). However group projects can be difficult to grade and there “are always ‘good’ students who contribute large amounts to a project while others do not spend enough time and effort and let others do all the work” (Chen, 2005). As a way around this Wells and Jones (2005) awarded individual marks to group members for their part in a group presentation and an essay and students were required to self-assess the value and quality of their work submitted.

Group projects were used by Nance (1998) in Systems Analysis and Design to improve students’ soft skills, e.g. team-work, group development and project management. The use of roles can help to develop the skills and understanding needed in work and provide a way of distributing workload (Reynolds, 2013). Wells and Jones (2005) observed that self-forming groups performed more cohesively as they were able
to draw on collective experience and knowledge from prior study. Mueller (2012) recognized that with five group members or above diminishing levels of motivation can be seen and the number of people who speak at any one time is harder to manage in a group of five or more. Research has also shown that the maturity and motivation of students plays a role in the success of group work. For example Rob (2006) found that Systems Analysis and Design teaching through group projects was found to be more suitable to graduates than undergraduates. And Shelly et al (2006) found that open ended case studies tend to be more suitable for graduates than undergraduates, undergraduates tending to work better with a defined case problem.

This research combines scaffolding with the benefits and shortcomings of case studies into an approach for assessing undergraduate Systems Analysis and Design students.

A Scaffolded Case Study Approach

The approach put forward in this paper was compiled through experiences of teaching Systems Analysis and Design over a number of years at a UK University. The Computing subject group at the university is part of the Business School and has four main degree pathways that students can study: Web Development, Games Design and Development, Business Information Technology, and Computing. Systems Analysis and Design is a mandatory module for all second year computing pathway students. Module content focused on Systems Analysis and Design from a structured and object-oriented perspective and also took both plan and agile development into consideration. Basic systems thinking was taught at the beginning of the module and students were encouraged to think in terms of a systems approach (Bertalanffy, 1968). Real life examples were used throughout to illustrate points, particularly from a business perspective. The module aimed to give students not only subject knowledge but also develop soft skills such as critical thinking, teamwork, communication and problem-solving.

The module was divided into two parts: part 1 which involved learning theoretical concepts and part 2 the application of theoretical concepts. Each part took place over a semester.

Assessment Part 1: Learning theoretical Concepts

Teaching for the first part of the module consisted of video lectures made available to students through mp4 and PowerPoint slides. This gave students the opportunity to review module material at any point during the module. Weekly seminars consisted of short practice exercises and time to complete parts of the assessment related to the lecture topics. Students were given set tasks to complete in order to build up their assessment week by week and weekly quizzes. Quizzes were formative but students worked in groups within a timeframe. This gave students an opportunity to test their knowledge of theoretical concepts from the previous weeks material and inform areas for further study. Topics for lecture and seminars in this part focused on learning theoretical Systems Analysis and Design concepts including systems thinking, the systems/software development lifecycle, agile and plan driven development together with structured and object oriented analysis and design using different modelling techniques. Figure 1 shows part 1 of the assessment.

- **Selection of predefined case study**
- **Tasks to complete for the chosen case study:**
  - Functional and non-functional requirements
  - Analysis and design of the new system using structured and/or object oriented modelling techniques
  - A process specification (e.g. structure chart, program flow, user interface designs with pseudocode)

Figure 1. Assessment part 1: learning theoretical concepts
The first part of the assessment (Figure 1) consisted of the selection of a predefined case study. Students could choose one of four case studies based around their computing degree pathway, e.g. online hotel booking system for the Web Development pathway. Students were given set tasks to complete for the case study, such as to write the functional and non-functional requirements, draw analysis and design models and produce a process specification. As a further support students were provided with a question and answer document which pre-empted questions that they might have about the assessment. This gave a more comprehensive explanation of tasks than those given on the assessment explanation document.

Although students could choose a case study from a different pathway to the one in which they were studying, students tended to stick to the case study which related most closely to their pathway studied. Students did not get the opportunity to develop/code the system, and the assessment was based around static modelling in order to understand core theoretical concepts before applying these to a real world context.

For the first occurrence of the module students were able to choose their own case study for this part of the module. This was to give them the freedom to be creative in an attempt to engage them in a topic area that was of interest to them. Students then used the same case study for their second assessment the group project (see assessment part 2). However even with tutorial support students tended to struggle to identify ideas and had difficulty scoping the case study as they were unfamiliar with Systems Analysis and Design concepts at this point. This was also reflected in student grades (Figure 3) and module evaluation results (see module results and evaluation section). Having a set case study provided further scaffolding (e.g. Bruner, 1966) for students for the first part of the module, helping them to develop a higher level of understanding through focusing their attention on learning the theoretical concepts and putting together the assessment tasks together with understanding what a case study is and contains.

**Assessment Part 2: Application of Theoretical Concepts**

Similar to part 1 teaching for the second part of the module consisted of video lectures made available to students through mp4 and PowerPoint slides. Topics focused on group work, conflict resolution, project management and planning, user interface design, testing, legal, ethical and social issues, critical evaluation and reflection.

This part of the assessment (Figure 2) involved group work in the form of systems analysis and design documentation, prototype development together with an individual element (40% of assessment grade). The module grade was therefore not totally reliant on the functioning of the group. Students used the seminars for group meetings and to work on their projects. The module tutors were available for guidance and spent time with each group on a weekly basis answering queries and checking progress.

**Group formation and role selection**

Student group size was set at 3 or 4 (e.g. Mueller, 2012). Students could select their own group but were advised to have a range of systems analysis, design and development skills within the group. Each student had to undertake roles (e.g. Reynolds, 2013) – which comprised one major role from system analyst, developer and project manager together with a minor role which could include a different major role or another role such as tester, researcher or designer. This enabled students to have a clear focus and expectation of what they should be contributing to the group. Each group submitted a form to the tutor with their group name, group members and roles so that a record could be made of each group and the balance of roles within the group checked.

Potential problems with group work were discussed at the start of the module so that students could recognise and deal with them in a professional manner. Students were encouraged to write down their concerns individually and share and discuss them with the rest of the group. Following this students devised a set of group rules to conform to during their project. Rules included ways of communicating, document sharing and version control, conflict resolution, etc.

**Project initiation**

Students chose their own real-life case study with a stakeholder as a way of encouraging engagement, deep learning, independent learning, creativeness and greater ownership of the project. As they had...
experience of what a case study is and contains (assessment part 1) and were encouraged to select a case study for a new system development or an existing system redevelopment which was relevant and of interest to them, such as a system needed by an employer for a part time job, students were more able and willing to engage.

Students worked through writing the aims, objectives and scope of their case study. This was signed off by a tutor to agree that it was feasible. The systems development approach/methodology was identified, e.g. agile, plan-driven, structured, object oriented and project risks identified with probability, impact and countermeasures stated.

Project management documentation was completed by students at an early stage and throughout the project in the form of a group gantt chart and individual group member gantt charts together with minutes of each meeting held covering who attended; what was discussed; what decisions were made; and any actions to be taken. Students also completed university ethical approval documentation which included the list of questions that they would use to elicit requirements from project stakeholders. This provided tutors with the opportunity to check student’s requirements elicitation techniques and questions and provide further advice/guidance where necessary. The preparation of both project management and ethical approval documentation also acted also as a practice learning experience for third year undergraduate dissertations whereby all students are expected to complete such documentation.

- **Group formation and role selection**
  - Discuss group member skills in systems analysis, design and development
  - Form groups of 3 or 4 members and select group name
  - Define group roles (one major and one minor role)
  - Discuss worries about group work
  - Set group rules

- **Project initiation**
  - Devise own case study and identify project stakeholder
  - Outline aims and scope of case study
  - Identify the systems analysis, design and development approach/methodology
  - Identify project risks
  - Project management, e.g. gantt charts, meeting minutes
  - Seek ethical approval

- **Analysis and communication of project plans**
  - Elicit and define functional and non-functional requirements and produce analysis models
  - Communicate project plans to tutors and peers

- **System design, build and test**
  - Produce design models of system and process specification
  - Develop prototype
  - Test prototype (black box, white box and unit testing)

- **Project evaluation**
  - Individual critical evaluation and reflection and grade peers within group
  - Showcase prototype to tutors, stakeholders and peers

Figure 2. Assessment part 2: application of theoretical concepts
Analysis and communication of project plans

Functional and non-functional requirements were elicited from the project stakeholder mostly through questionnaire and/or interviews and analysis models devised. Students were encouraged to use MoSCoW for requirements prioritization and give their list of requirements to a group of peers to comment on to encourage the elaboration and thorough definition of requirements.

In week 4 student groups communicated their project plans to tutors and their peers in the form of a presentation. This comprised of the roles of group members, project aims, the intended programming language, project risks, systems analysis/design approach/methodology, ‘must have’ requirements and an example of a non-functional requirement. This helped to encourage students to refine project ideas and provide motivation for completing work to date. Tutors were able to check whether ‘must have’ requirements were within the scope of the module timeframe and level of difficulty for the module.

System design, build and test

Similar to Assessment part 1 students were tasked with producing design models and a process specification (including structure chart, program flow, user interface designs and pseudocode). This part of the assignment was distinctively different to part 1 as it required students to build and test a prototype of their case study system. This required the use of a programming language to practice skills gained in other modules. Black and white box testing was utilized together with unit testing throughout prototype development.

Project evaluation

The assessment also included an individual element in the form of a critical evaluation and reflection. Students also had the opportunity to peer review the performance of others in their group. The grade given by the students in the peer review gave the marking tutor an indication of the level of effort of each student within the group. Students presented their group work at a ‘showcase’ event whereby module tutors marked the prototype of the computer system developed. Other students, lecturers and project stakeholders had the opportunity to view students work during this event and ask questions.

Module Results and Feedback

Figure 3 shows a comparison of overall module grades for module occurrence one (where students selected their own case study for part 1 of the assessment) and module occurrence two (use of a predefined case study for assessment part 2). The spread of grades on modules one and two is clearly evident in figure 3 showing a peak in C grades for module one and a peak in B grades for module two. 64% of students on module two achieved either an A or B grade compared to 40% of students on module one. The percentage of failure grades (E to H) reduces from 14% for module one to 2% for module two.

An evaluative questionnaire was given to students to complete anonymously at the end of each module. The questionnaire comprised of fifteen questions relating to the availability of resources, approachability of the module tutor and organisation and engagement of the module. The first occurrence of the module which enabled students to select their own case study for assessment framework part 1 received a 43% response rate whereas 62% of students completed the evaluation for the second occurrence of the module with a predefined case study. For module occurrence one 70% of students said that they would recommend the module to others in comparison to 85% of students on module occurrence two. 55% of students felt fully engaged with module occurrence one in comparison to 88% of students studying module occurrence two.

67% of students on module occurrence one indicated that the module would enable them to develop skills that would help their employability or career development in contrast to 88% on module occurrence two. Indeed group work can assist with ability to interact with peers on different levels, to take responsibility for their own learning and develop responsibility for the learning of others in their group (Wells and Jones, 2005). 67% of students on module occurrence one also stated that they learnt a lot from the module compared to a higher figure of 83% for module occurrence two. 82% of students on module occurrence one indicated that they were provided with helpful feedback on their progress within the module (e.g. question and answer sessions, tutorials, email communications, feedback on assessments, etc.) compared to 95% of students on module occurrence two. 69% of students on module occurrence one
found the assessments stimulated their learning and developed their skills and knowledge compared to 82% of students studying module occurrence two.

![Bar chart showing grades comparison](chart.png)

**Figure 3. Comparison of module grades: own case study selection (module occurrence one) compared to predefined case study (module occurrence two) for assessment part 1**

Student feedback for assessment part 1 included the following comments: ‘I love the fact that we have video lectures so we can concentrate on doing exercises in class’; ‘the activities in class are interesting and engaging’; ‘the case study enabled us to get to grips with core concepts’.

The overall standard of group work projects produced was extremely high. There was a ‘buzz’ and enthusiasm shown during the showcase event and students were proud to ‘show off’ and explain their work. Being able to practically develop/code the design helped students to visualise the design and reflect on whether the analysis in the form of requirements and associated models were adequate and met stakeholder needs.

The freedom to create their own case study in assessment part 2 gave them ownership of their project and resulted in high overall module achievement. Also the case studies had a dynamic element as they were based on real life projects. Students were also focused and driven in the seminars when working in groups. Group members pushed one another to achieve for the group effort.

**Conclusions**

It is important to make Systems Analysis and Design teaching and learning engaging and relevant to students and emphasise its importance as a foundation for systems development. Giving students the opportunity to devise their own case study can help with engagement, deep learning, enthusiasm and encourage independent study within a discipline that can be challenging for both teaching staff and students. Although Rob (2006) advocates the use of group projects for graduates in Systems Analysis and Design teaching rather than undergraduates, when group project work is supported through tutor and peer review together with learning theoretical concepts and the use of a defined case study problem (e.g. Shelly et al, 2006) prior to group work effective results can be ascertained. Furthermore the integration of scaffolded supports and learning and applying theoretical concepts through prototype construction can help to reinforce principles fundamental to the analysis and design of systems.
References


IT Jobs Watch 2017. *IT Jobs Watch, Tracking the IT Job Market* [online] Available from: [http://www.itjobswatch.co.uk](http://www.itjobswatch.co.uk) [Date accessed: 15 February 2017].


