Co-design and Co-deployment Methodologies for Innovative m-Learning Systems

David Millard, Yvonne Howard, Lester Gilbert and Gary Wills
Learning Societies Lab,
School of Electronics and Computer Science
University of Southampton

Abstract

Building innovative m-learning systems can be challenging, because innovative technology is tied to innovative practice, and thus the design process needs to consider the social and professional context in which a technology is to be deployed. In this chapter we describe a methodology for co-design in m-learning, which includes stakeholders from the domain in the technology design team. Through a case study of a project to support nurses on placement, we show that *co-design* should be accompanied by *co-deployment* in order to manage the reception and eventual acceptance of new technology in a particular environment. We present both our co-design and co-deployment methodologies, and describe the techniques that are applicable at each stage.

Keywords: Co-design, Co-deployment, Agile Development

Introduction

In the last decade we have seen many types of mobile (m-) learning tools, from simple systems that allow access to existing content and functionality on-the-move (Flynn *et al.*, 2000, Collins, 2005) to more targeted applications, that take advantage of the mobility or locality of the applications users, for example, to provide location-based information (Abowd *et al.*, 1997, de Crom, 2005), or to support fieldtrips (Kravcik *et al.*, 2004, Weal *et al,* 2006). However, these tools often replicate existing learning activities, rather than changing the nature of the activity itself.

This is understandable as changing practise also requires a change in philosophy or of culture, and is far more difficult to achieve even in small problem areas. It requires an appreciation of the problem space that is usually beyond the technology experts, and an understanding of the technology that is usually beyond the domain experts. This creates a tension in the design process as no single expert has the necessary knowledge or skills.

Design methodologies are therefore required to drive the creation of novel tools, ensuring that they are both useful and sustainable in practice. Many methodologies and models for the design of information and e-learning systems take a layered approach, separating design issues to allow independence (Wills *et al.*, 2003): mapping the domain (in terms of its structure, content, work flow, *etc)*, analysing the associations and relations in that domain, and presenting the information to appropriate users.

Our co-design approach has the objective of ensuring that an m-learning system is both feasible and useful by explicitly including the expertise of the people in the intended domain, and as such it is similar to the socio-cognitive engineering approach proposed by Sharples *et al.* (2002), which seeks to develop a theory-based framework of the user's underlying cognitive and social processes. Our co-design methodology is more lightweight, and does not attempt to build a theoretical model of users' practice, but instead works closely with users as part of the design team in an agile way—something that is essential for tools that innovate practice. Our methodology involves domain experts as 'first-class' members of the design team, using a number of techniques that brings the design team together and helps it to converge on a joint understanding of the conceptual space, and focusing on tools to tackle real problems in the domain. We have done this by integrating techniques found in HCI (personas, scenarios, and storyboarding) with agile software development techniques (iterative and incremental delivery) and lightweight software engineering (use case, activity, and iteration diagrams).

We have also discovered that the engagement with the eventual user group must continue into the deployment phase of prototyping, using co-deployment methodologies that emphasise a continuing conversation between a range of stakeholders in the user community and the design team.

In the rest of this chapter we present our co-design and co-deployment methodologies, using a case study to demonstrate how some of the techniques work, and showing the consequences of underestimating the importance of the deployment phase. Our hope is that these methodologies will help other m-learning developers to design new technology and applications that create genuine innovations in the domains in which they are deployed.

Background: Design Methodologies

Our work on co-design and co-deployment builds on traditional software engineering practices for m-learning, and especially on participatory design and agile development. Co-design and co-deployment are focused on enabling innovation in a domain through the use of technology, and as such can also benefit from the experiences of management methods for institutional change.

Traditional Software Engineering

Mobile learning development has often followed traditional software engineering methodologies where teams of developers envision, implement, and deploy systems. In so doing, mobile learning systems are not immune from the software engineering paradox described by Lehman (1980) satisfaction declines unless steps are taken to constantly improve systems. This effect can be detected even before a system has been completed, so that when a system has been completed to an agreed specification, on deployment, users feel dissatisfied with it. Lehmann suggests that the activity of envisioning a system creates technology transfer from the technology specialist to the domain experts, increasing the domain's awareness of opportunities for change and its expectations for the deployed system beyond the originally agreed goals.

Efforts to mitigate these effects have included agile development with its principles of domain expert involvement in the design team throughout the design phase, and participatory design methods where users are first class members of the design team.

In mobile learning, traditional software engineering methodologies have been challenged to adapt to the specific needs of designing for the mobile learning experience. Parsons *et al.* (2006) examined m-learning in particular, and identified four design concerns:

- **generic mobile environment issues** – such as communication support and the device interface
- **learning context** – including the roles of users, the collaborations and activities
- **learning experiences** – structuring the learning, e.g. through cinematic or game metaphors
- **learning objectives** – the desired goal, for example improved skills or social abilities

Participatory Design

Participatory design is a long established practice in Human-Computer-Interaction (Grudin and Pruitt, 2002). It challenges the traditional view that after the requirements gathering/ elicitation stage, end users are not required and that they should let the 'experts' get on with the job and design the system. Participatory design brings the end users into the design team as equal participant (members) of the design process. For instance, participatory design has been used in the design of mobile systems (Svanaes and Seland, 2004); and Massimi *et al.* (2007) have used participatory design to design mobile phones for the elderly.

Other approaches have also been developed that include users in the later stages of design. In their work on m-learning, Sharples *et al.* (2002) developed a socio-cognitive engineering approach, a theory-based framework of the user's underlying cognitive and social processes. Socio-cognitive engineering is wary of user-centred design as users are not always able to articulate their own working pattern and methods; instead the method seeks to develop a theory-based framework of users' underlying cognitive and social processes. In practical terms this requires two studies: an investigation into how user activities are performed in their normal contexts, and a theoretical study of the underlying cognitive and social processes.

Agile Development

Agile methods are a number of software development methods that were proposed in the mid 1990s as a reaction to traditional approaches. An agile method could be defined as an adaptive process run by talented and creative people and controlled with iterative and incremental development (Abbas *et al,.* 2008). Although agile methods were initially described as development methodologies, the term *agile* represents an attitude, a philosophy, and a way of thinking that was presented through the principles and practices in the agile manifesto (Highsmith *et al.,* 2001). This way of thinking can be applied to many other aspects of software creation including design and modelling. Agile techniques share common principles such as (Larman, 2004):

- delivering working software usually within a short timescale
- close communication
- simplicity
- preferring programming over documenting
- customer involvement
- encouraging rapid and flexible response to change.

We advocate agile methodologies for co-design and co-deployment, as these are the most transparent to non-technical team members, and also can react quickly to changes in context or circumstance.

Change Management

Our approach to co-design is intended to encourage innovation and change of practice. Introducing technology usually changes current practice, at a minimum changing the standard operating procedures. However, introduction of technology also has a wider effect than just the immediate application, department, or division (Yusop *et al*., 2005). Hence the change has to be managed carefully. This is especially true in the case of co-design, which encourages the team to not only think about implementing a solution, but to re-examine underlying assumptions and practices, encouraging innovation and change in practice.

The process of change needs to be managed with care to ensure that all stakeholder are positively engaged, especially those who have the power to implement the change (primary stakeholders), and those who have influence over opinion within the organization. Hence it is essential to carry out a full stakeholder analysis. As with any change management, when it comes to implementing the change it is important to identify champions in each of the stakeholder groups, coupled with clear and regular communication.

Co-design Methodology

An overview of the five stages in our co-design process is shown in **Error! Reference source not found.**. Each stage is supported through workshops and design meetings, attended by both the technical and domain experts in the design team. While there is a natural flow from stage one through to stage five, the design methodology is a natural cycle involving feedback from later to earlier stages, and design teams will typically undertake several iterations of the last three stages. Each stage, and the techniques used in that stage, is explained in the following sub-sections.

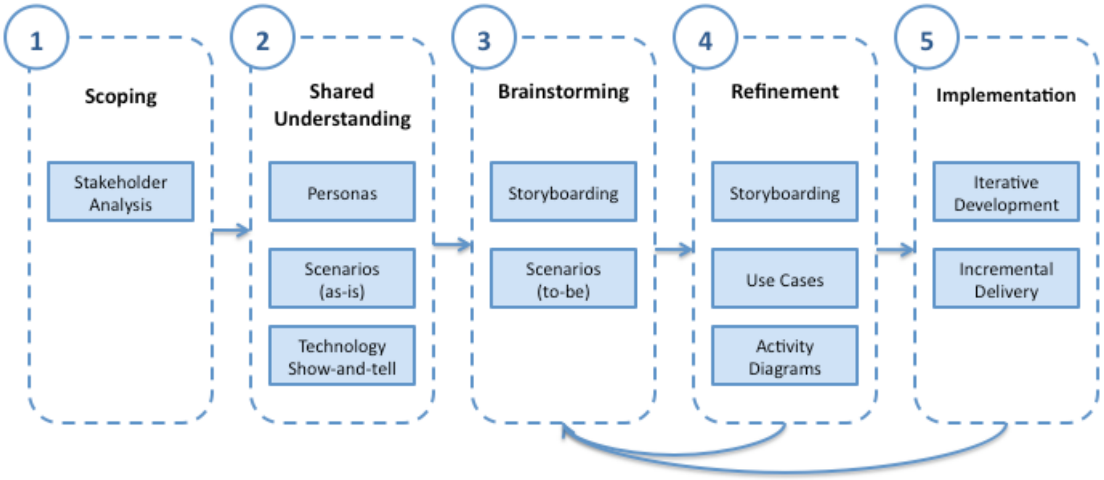


Figure 1: Overview of Agile Co-design Methodology

For clarity we have not shown the usual project quality reviews and documentation that takes place between each stage. Similarly with the development of code and in-line with good practice, unit tests and integration tests were written before the code was produced.

Scoping - Primary Stakeholder Scoping (Establishing Domain Partner)

Before co-design as such can begin, there is a stage where the co-design itself is defined and planned. The first priority is to establish the domain and technology partnerships in the venture from which the management team will be drawn.

The objectives of the planning team are to ensure that co-design principles are observed and that the design outcomes are achieved. It defines the scope of activity and identifies co-design goals. It also identifies, selects, and recruits co-design participants, identifies, articulates and shares common goals and purposes with co design participants, and defines and carries out the co-design plan to fit overall delivery iterations.

Stakeholder Analysis. One of the first activities of the planning team is to select and recruit the co-designers using stakeholder analysis to ensure that the domain within scope is fully explored. There are a number of stakeholder analysis models, but the model we have chosen (Dix *et al.*, 1993) divides stakeholders into four categories:

- Primary stakeholders: people who use the system directly
- Secondary stakeholders: people who do not use the system directly but receive output from it or provide input to it (indirect users)
- Tertiary Stakeholders: people who do not interact with the system either directly or indirectly but who are affected by its success (or failure)
- Facilitating Stakeholders: people who are involved with the design, development and maintenance of the system.

Having identified and categorized the domain stakeholders in scope, each stakeholder's concerns and characteristics are defined and their influence and impact in the domain is assessed. When these aspects are fully understood, then the co design team is more likely to achieve its goals.

The planning team selects the most important stakeholders from each category, and recruits representatives to join the co-design participants. The planning team should aim to recruit a group of between six to eight co–designers with a focus on those primary, secondary, and facilitating stakeholders who are closely identified with the needs of the primary users in the domain scope. The planning team's expertise and experience is crucial in identifying potential co-design participants who have the enthusiasm and commitment to achieve the co-design goals.

Shared Understanding (of Problem domain)

Once the stakeholders have been identified, and have agreed to come into the team, the next priority is to create a *Shared Understanding* of the design space. Typically, this means that technicians learn about the values of the application domain, and that domain experts are introduced to the technology, its scope, potential, and limitations. The intention is not to bring everyone in the team to the same level of expertise (this is not possible in sophisticated domains or with sophisticated technologies) but to enable an informed conversation to occur in the other stages.

Personas and scenarios are a lightweight method for capturing and recording the requirements of a system from an end user's viewpoint (Cooper and Reimann, 2003). A persona describes an end user in some detail; their background, job function, and situation in the organization. Scenarios are textual descriptions of how a persona interacts with the system and other personas when using a system. The scenarios are independent of any technology and they may represent either current practice (as-is) or improved practice (to-be). In this stage they will typically reflect the domain 'as-is'.

Technology Show-and-Tell. It is important that all members of the design team are familiar with both the current technology being used in the domain, and with new technologies that could create new opportunities. We have found that a show-and-tell event is a useful way of bringing developers and users together (often for the first time), breaking down social barriers, and creating a common vocabulary for the design team to move forward. Sometimes developers will have to visit key locations to understand the context of technology use, but usually a workshop can be held that is focused around a number of key props. Props can include pen and paper based systems, as well as electronic and computing devices.

Brainstorming

The next stage involves *Brainstorming* ideas for new applications and tools. It relies on the shared understanding established in stage two, and builds a number of initial design artefacts based on the common vocabulary, in particular new scenarios and storyboards of potential applications.

Scenarios (to-be) and Story Boarding. A second set of 'to-be' scenarios can be written to capture ideas created by the group. These are textual descriptions that can draw on the same personas used in the earlier stage. The 'to-be' scenarios describe how the personas might interact with potential applications to fulfil existing needs (or, for innovating developments, new needs). Using the scenarios, story boards can be created to represent the user interface design (UI) of a given tool. This is a standard technique used in HCI development and is very effective when used in a participatory (or co-design) process. Both the end users and developers (HCI experts) are engaged with designing the UI. During this process, the scenarios can be clarified and modified if required.

Refinement

The next stage is one of *Refinement*, where the informal ideas captured in stage three are converted into a more concrete set of software requirements and specifications. We take an agile approach, and use lightweight documentation as a means to drive development, rather than as a passive record of activity. We have found three formalisms particularly useful: ontological modelling, use cases, and activity diagrams.

Ontological Modelling. Identifying key resources and mapping the relationships between them is a significant part of any co-design process involving conceptual spaces. Often domain experts will not be aware of what types of structure exist within the conceptual space of their domain (for example, is it a taxonomy or just a hierarchy, do richer relationships exist, if so what are the constraints?). Key information structures may have evolved rather chaotically, and modelling them may be a useful point of reflection for domain experts. Ontological modelling is expressive, makes no assumptions about the underlying information models of the domain, and can be easily communicated to domain experts in the form of entity-relationship diagrams.

UML Use Cases and Activity Diagrams. Use Cases are an excellent high-level (and implementation independent) starting point for describing functionality in the context of a given system and user. We use standard UML 2.0 use cases, consisting of a use case diagram, with success scenarios for each case. A brief narrative description is held alongside the diagram as a whole, as well as for each individual use case. From an agile point of view they are effective because they are relatively informal, yet help to define and capture a problem space in detail that can be understood by the whole team, including the end users.

Implementation

The fifth and final stage is *Implementation*, where the design artefacts created in stages three and four are used to drive the creation of the application or tools. We use agile software engineering practices based around iterative development and incremental deployment, and assume that other activities key to the software engineering process (such as architecture design, testing and evaluation) are consistent with these principles.

Iterative Development and Incremental Deployment. Agile methods are a number of software development methods which were proposed in the mid 1990s as a reaction to the limitations of traditional software development methodologies. Although these methods vary in practice, they share common principles, such as delivering working software frequently within a short timescale, close communication within the team and with the customer, simplicity, and programming over documenting (Larman, 2004). We have applied the spirit of these principles to the earlier stages of co-design, but they are especially important in the final implementation stage. Development should be focused in a number of small iterations, with the design team (including the users) reviewing the progress of each iteration in a design meeting. The development is shadowed by incremental deployment of the application, which first engages users with a simple (but stable) application from early iterations, and then gradually introduces new functionality. In some cases this will require the team to revisit earlier stages, to reassess assumptions and revise requirements.

Case Study: mPLAT

In this section we present a case study of using this agile co-design methodology to help create an m-learning tool for nursing students on placement. The Mobile Placement Learning and Assessment Toolkit (mPLAT) project aimed to provide a mobile learning toolkit to support practice based learning, mentoring, and assessment to these nursing students. Sloan and Delahoussaye (2003) have shown that nurses benefit from mobile access to information, and the mPLAT system applies this idea to student nurses. Our belief going into the co-design process was that practice-based learning and the mentoring process would be improved with tools that connected the student *in situ* with the competency model against which they were being assessed, and were required to learn.

Motivation

The following scenario illustrates the problems and the need for such a toolkit:

Pre-registration nursing students spend 50% of their 3-year programme in clinical practice undertaking a series of placements in different areas of the healthcare system. Mentors support students for the duration of their placement. Mentors assess the students' competence in practice against a set of learning outcomes detailed in the practice assessment booklet or practice portfolio. These are summative assessments which students are required to pass in order to register as a nurse at the end of their programme. Students are expected to complete a preliminary, an interim and a final interview with their mentor. The interim interview is crucial as it is at this point that the student who is failing to progress is likely to be identified and action plans can be put into place. This good practice feature of induction, interim, and final assessment is common to most educational situations where students experience work-based learning situations.

Issues around ensuring that students are fit for practice at the point of registration were brought home recently following a report by Duffy (2004) which found that mentors were reluctant to fail students due to a number of factors, including lack of confidence, concerns over personal consequences (for student and self), and leaving it too late to implement formal procedures (the preliminary and interim interviews missed or undertaken so late that action plans to assist a student who is not progressing cannot be implemented).

The Co-Design Process

The co-design process was initiated with a number of workshops. We invited domain experts and stakeholders to join the project team. The main purpose of the first session was to acquaint the co-design team with our individual expert areas, and to allow us to exchange knowledge so that we built a shared understanding of the domain and technologies in preparation for the following two co-design sessions. The second co-design workshop focused on a brainstorming exercise to facilitate thinking about small novel applications that could help solve some of the problems identified in the first workshop. The third co-design workshop was a refinement process, where we selected three candidate applications and explored their requirements in more detail. It became clear at an early stage that the main concern of the nursing team was in trying to connect student's practice with the conceptual competency framework used by the School of Nursing. As a result we identified a number of potential tools based around this concern:

1. **Profile Placement Tool** would provide guidance for mapping the domain and competencies to the experiences (opportunities) offered in the placement area (e.g. care delivery in a medical ward).
2. **Learning Contract Builder** would draw on the student's and mentor's experience and the placement profile to create an action plan concerning what the student will achieve, how they will achieve it, what evidence is required, and which learning resources are appropriate.

Personas and Scenarios

Personas and scenarios were actually written by the end users, with a little guidance. In addition to the different institutions, there were also the requirements from the professional body (Nursing and Midwifery Council) and the British National Health Service (NHS) to capture, so we developed a number of personas and scenarios for each role. Student surveys have identified the character of a good and poor mentor (Gray and Smith, 2000). A summary of three of our personas for the mentors were:

- The ‘gold-standard’ mentor: one who facilitates learning appropriate to the student’s level of ability and makes an appropriate assessment of the ability.
- The ‘toxic’ mentor: The term ‘toxic mentor’ was initially coined by Darling (1984), but discussions with mentors and students currently in practice indicate that this is still relevant. Darling described a gallery of toxic mentors as Avoiders, Dumpers and Blockers.
- The ‘hero’ mentor: the mentor who refuses additional help with a student despite potential difficulties such as personality clashes, professional value issues, etc. It appears that these mentors believe that they can turn anyone into a nurse and that if they don’t it is their fault and not the student’s.

These were accompanied by scenarios describing the situations in detail and the problems mentors faced while trying to carryout this function and still perform their duties on the ward. These personas and scenarios were developed from a combination of a literature review and interviews with current mentors and student advisors. Those listed above aided the first set of co-design workshops and helped the technical members of the design team become familiar with the issues faced by the end users.

Storyboarding and Ontological Modelling

From the co-design session, personas, scenarios, and the competency profile of the student user, we were able to create storyboards for the tools quite quickly. We first identified the key features from the scenario and sketched out the initial ideas. In parallel with the storyboarding we were developing our use case and activity diagrams. Our preliminary storyboard presented the basic ideas regarding the user interface, for example using tabs to allow users to recognize the available functionality.

We also explored the competency model that lay behind the portfolio (a simplified overview of this is shown in Figure 2). Through the modelling process we discovered that certain terms used were ambiguous or overlapping, which seems to have occurred as a result of combining several other competency models from professional bodies, the UK government, and the University’s School of Nursing.

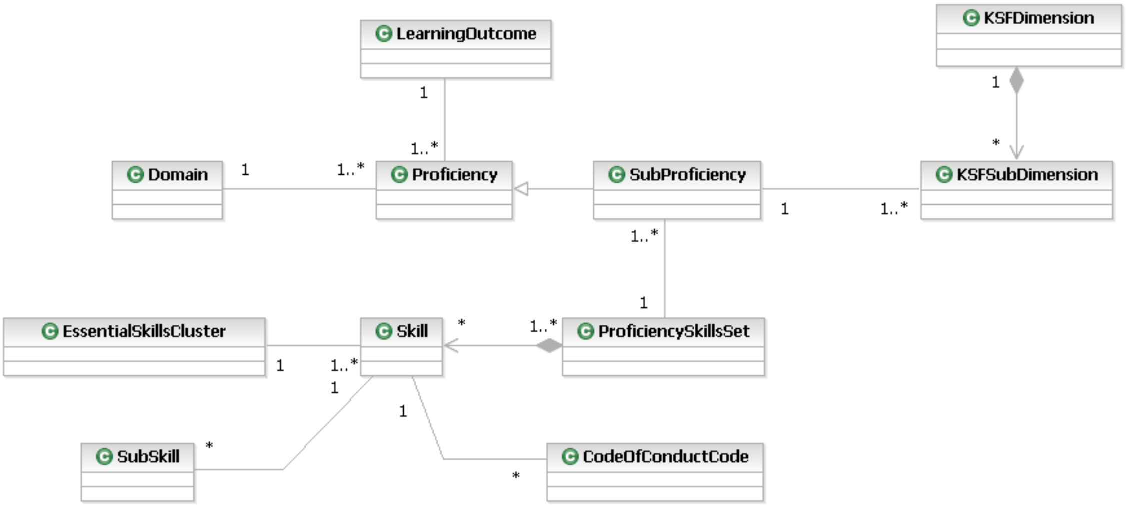


Figure 2: Initial competency model identified through ontological modelling

Use cases and Activity Diagrams

The development of the use case and activity diagrams was again a co-design exercise but on a small scale; just one or two members of the Nursing team joined in the activity. Figure 3 shows the use case diagram for the system. This was supported by normal use and alternative use scenarios for each use case.

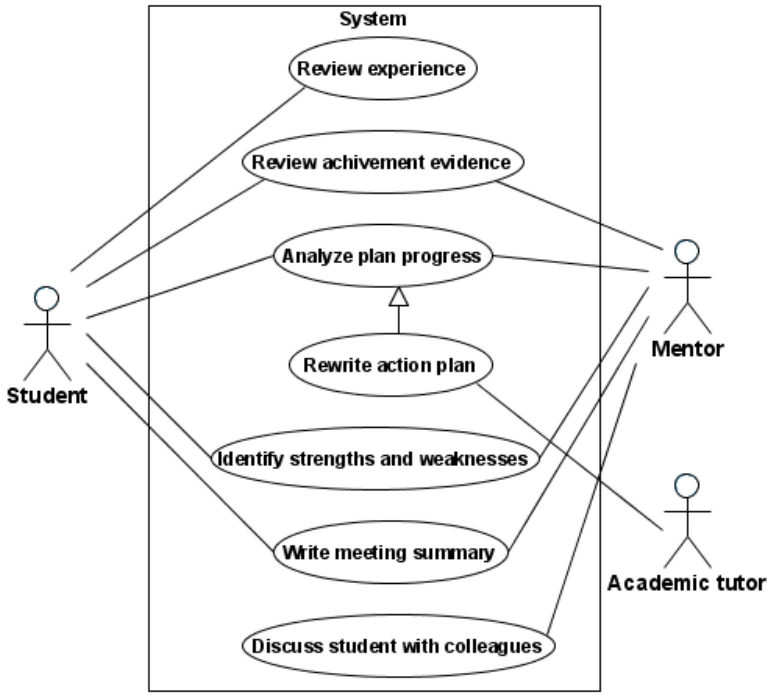


Figure 3: Use case for the system

When supporting students to assess their competency for a task it was necessary to decide how much 'scaffolding' support was required. Should we build in a very structured approach, directing them in the way they should go, or a looser approach and let them find out themselves? We decided to adopt something between the two extremes, with a leaning toward the more structured approach. The reasons for this were:

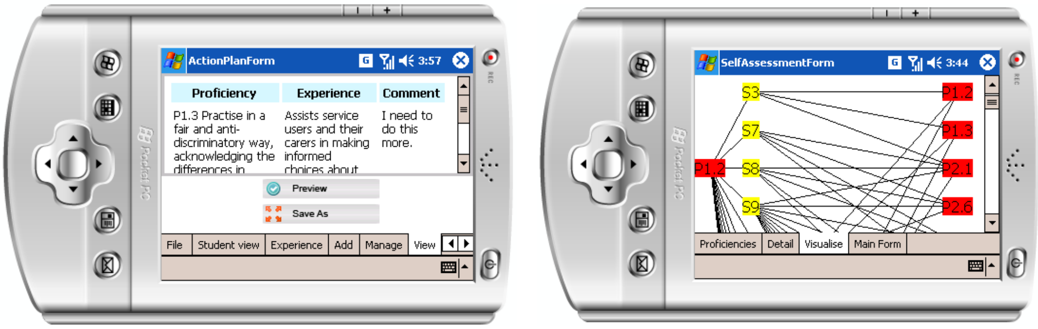
- Part of the learning process involved students becoming reflective practitioners; being too prescriptive would not give students the opportunities they needed to self-assess properly.
- There was value to a degree of structure, provided students were allowed to reflect on their learning. Giving them some structure would aid them to start their reflection.

As this was a new tool, and the entire system of practice assessment had changed, it was felt that a structured approach would make it easier for users to understand and use. A less structured approach was the goal, but was not realistic for an initial implementation. Rather than just providing a tool that mimics the multiple views of the current paper-based competency system, we wanted to provide functionality that supported the preparation for the initial meeting wherein the Action Plan is co-created by the student and mentor.

Implementation

The self assessments, placement profiles, and action plan applications were developed in .NET for the Windows Mobile platform. We developed from the start with the mobile platform in mind, i.e. the less-featured platform first. We developed separate projects under a Visual Studio 'solution' which represent different areas of programming focus, for example, database functions, data objects, GUI, and file I/O. Because of the limited storage of mobile devices, our code was combined into one project when deployed to a mobile device.

One of the essential uses of the tool was for students to be able to understand the competence model and how it was applied to their work placement. The Learning Contract Tool gave an overview of the competency model, which included a graphical view of the competency network and the student's progress within it, as shown in Figure 4.



Proficiency	Experience	Comment
P1.3 Practise in a fair and anti-discriminatory way, acknowledging the differences in	Assists service users and their carers in making informed choices about	I need to do this more.

Figure 4: Mobile application showing an Action Plan (left) and Competency Network (right)

Experience of Deployment

In mPLAT we used the co-design methodology but did not at first extend the principles into the deployment phase. Instead we planned and carried out a careful, traditional deployment. Nursing students and their mentors undertook a six-week placement in a clinical ward setting supported by the mPLAT tools. The deployment was supported by a domain expert who had been a member of the co-design team, shared the vision of project, and understood both the set of tools and the users. Student nurses volunteered to join the trial. We planned two-stage training for them, firstly in using the mobile device and its native tools (calendar, phone, email etc.) and then in the mPLAT tools. We set up a helpdesk and website that provided guidance and training materials for the users to browse or download. We informed the deployment locations (clinical wards in hospitals) that some of their placement students were taking part in a mobile learning trial that would involve the use of mobile devices in ward settings.

All of this preparation should have ensured a successful trial that would have enabled us to evaluate the goals of the project: Had student's holistic understanding of nursing proficiencies and skills improved? Were mentors more empowered to assess students? Were students more able to remain connected to their learning environment?

Unfortunately, the trial did not go according to our plans. Many of the students gave up using the mobile tools within the first few weeks of deployment, and by the end of the trial there were few users.

Using focus group meetings with the student nurses to find out what went wrong, we heard reports of ward staff demanding that the mobile devices were 'put away' immediately, and that they were not be used in a ward environment. Some student nurses were accused of 'texting their friends instead of working'. Some students found the mobile device itself technically challenging, and with the generally negative atmosphere did not feel motivated to overcome the difficulties. This happened even though domain partners were sponsors of the innovation and even though the deployment areas were informed about the trial.

Unfortunately we had overlooked some important truths: the technology and tools that we were deploying were more than beneficial innovations in placement practice; they represented change to a community with an identifiable and strong culture of norms, practices, and processes. Our student nurses, who were spearheading this innovative practice, were the least powerful members of the community we were deploying into. (It should be noted that these tools were new for nurses but similar devices were already established for other professionals working in the same environment.)

In our focus groups, we were able to find examples where determined students made good and effective use of the tools in their practice: looking up web resources to support their studies, and making journal entries in the placement context to help them write up their portfolios later.

This sobering experience helped us to recognize that co-design for creating useful, innovative tools is not enough to ensure innovation in practice; it requires a method that we are calling *co-deployment* which recognizes the difficulties of deploying tools which may challenge long-held practice, creates initiatives to mitigate them, and brings all stakeholders in the domain community to work together to accept beneficial innovation.

In the second mPLAT trial, we have been developing a co-deployment methodology that attempts to meet some of these challenges. Already we have improvements in domain community understanding of the scope for innovation and a lowering of some of the barriers to change. Co-deployment has brought together a team with a shared understanding of the benefits of innovation and a commitment to make the deployment successful through working with the community stakeholders. Through conversation and co-operation, barriers to change are being uncovered and strategies to overcome them developed.

Co-deployment Methodology

Co-design is focused on producing innovative applications, which leverage new technologies to create beneficial changes in practice. However in the deployment phase these changes are likely to be resisted, regardless of their value, as people working in a domain are already invested in their existing processes and methods, and may have suspicions about new technology and the motivations behind its introduction.

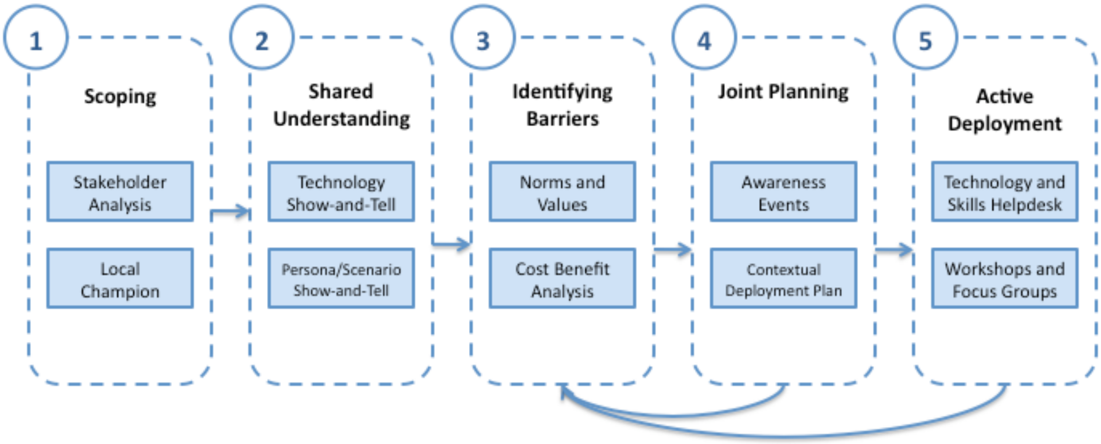


Figure 5: Stages of Co-deployment

Co-deployment is the process of involving the domain community in the gradual deployment, evaluation and revision of the m-learning application. The goals are to create community awareness, engage with existing groups, build skills, and enable a conversation between the development team and the domain stakeholders. It is based on the notion that any m-learning deployment has secondary and tertiary stakeholders, who must also be managed if innovation is to occur within a particular setting. Co-deployment has the same general shape as co-design, but is more focused on a particular context. It should start and run alongside the Refinement and Implementation stages of co-design.

Figure 5 shows the stages of co-deployment, which are further explained in the following subsections.

Scoping

The planning team has similar responsibilities in the co-deployment phase as in the co-design phase of establishing the co-deployment team and guiding its activities to a successful outcome.

Stakeholder Analysis. We use stakeholder analysis to identify potential participants for the co-deployment sessions. However, here the focus shifts to the specific deployment context which may be different than the general co-design context. The stakeholder analysis is extended to identify co-deployment team members for use in planning for active deployment. The set of tertiary stakeholders is expanded to include people in the deployment community whose concerns and attitudes may influence deployment success. Once again the concerns, characteristics, and spheres of influence of each stakeholder are assessed together with another aspect: their power within the deployment community.

Representatives from each category (but not necessarily from the expanded set of tertiary stakeholders) are invited to join the co-deployment team, again using the experience of the planning team to primarily select individuals who will have the enthusiasm and commitment to achieve a successful deployment. Since one of the goals of using co-design is to innovate practice using new technologies, there are likely to be groups of individuals to whom such changes initially seem threatening. The planning team should seek out influential representatives of tertiary stakeholders to join the co-deployment team, members who understand and may even share these concerns and who are also influential representatives of their stakeholder group. The other tertiary stakeholders are later addressed in stage 5, active deployment.

Local Champion. At the same time as establishing the co-deployment team from the stakeholder analysis, the planning team should recruit a project support person or 'champion' who will not only support users and the deployment community during deployment, but will champion the goals of the deployment with stakeholders. The support person is a first class member of the co-deployment team.

Shared Understanding

The early sessions of co-deployment are very similar to those of co-design. Once appropriate representatives from the stakeholder groups have agreed to join the team, the first priority is to build a shared understanding of the deployment space. The technologists learn about the practices, processes, and behavioural norms of the deployment domain culture, and also the practical limitations of schedule and timing. The domain stakeholders learn about the technology, its potential for innovation in practice, and its limitations.

Co-deployment uses similar techniques to create shared understanding of the deployment context:

Technology Show-and-Tell. We have found that a show and tell event helps to create a shared understanding and helps the domain community members to quickly feel familiar with the technologies and tools being deployed. The event demonstrates the tools, applications, and technology that will be deployed, showing potential and scope but also limitations. Deployment community team members describe the environment and its physical and cultural characteristics as the context for deployment. The event helps to break down social barriers within the team and create a shared vocabulary.

Persona and scenario show and tell. In a similar way to co-design, personas and scenarios are a useful way of exploring technology in a deployment context, creating narratives of use with recognizable personas. Personas can be derived from both co-design and co-deployment stakeholder analyses describing their motivations and behaviour in the deployment context. Scenarios should be firmly situated in the deployment context.

Identifying Barriers

There are going to be barriers when introducing technology into an organization. Stage three of co-deployment involves identifying these barriers and knowing which are the ones that can be worked with, although possibly limiting the application, and which are the ones that should be challenged.

Within a given working domain or context, there is generally a difference between the designers norms and values and those of the users of the technology. This is in addition to the recognized barriers to the introduction of technology: senior management commitment, buy-in from users, use of technology that has been fully tested, good communication, *etc*. (Brown *et al,*. 2007). Resistance to change can come from many quarters when introducing mobile technology. The specific issues are:

- Intrusion; mobile devices can be seen as an intrusion into current work practice.
- Privacy; many of these devices can capture audio and video, and if used unprofessionally may compromise a client's privacy.
- Digital divide; we now have the situation of digital natives and digital immigrants (Prensky, 2001), that can threaten the hierarchies and social norms in the work place.

This stage is concerned with identifying *Barriers to Change*, and the *Barriers to Challenge*. We use two different techniques to achieve this:

Norms and Values (Identifying Barriers to Change). The values of a community are often related to their culture and belief system (so for example, in nursing, respect for the person and caring for their needs are very strong values). The norms are closely related to values, concerned with the ways the community normally works (professionalism), and the implicit structures that exist within the community **(**Kling, 1996). Value Exchange recognizes that people will do things if it is of value to them or they get something of value in return. For instance, people will take the time to learn a new system for many reasons not directly related to money (Gordijn and Akkermans, 2001): they may want to make the task easier, want to be seen to be in the 'know' or up to date, believe it will help the patient for whom they are working for, etc. We use semi-open interviews and small forums to try and identify norms and values.

Cost Benefit Analysis (Identifying Barriers to Challenge). CBA is a group of techniques that assign a monetary value to activities or artefacts in order to ascertain if a project is financially worth starting or continuing with, indicating when and at what rate the return on investment will be realised. We seldom do a full and formal CBA, but the process can be useful in identifying which of the norms may be worth challenging, and which are linked to values that may not be apparent to developers from a different working culture.

Joint Planning

The co-deployment team, having established a shared understanding of the opportunities afforded by the new technologies and also of the difficulties of deploying the technology in practice, need to establish a context for the deployment.

Awareness Events. In this activity, the concerns of secondary and tertiary stakeholders are addressed by creating awareness through workshops, presentations, demonstrations, and interviews. The deployment community is given the opportunity to voice their concerns and the co-deployment team can demonstrate where and how those concerns have been addressed, and also dispel misunderstandings. The co-deployment team needs to be open about the deployment and demonstrate their willingness to both listen and respond to legitimate concerns, be robust in dispelling misunderstandings, but do so in a manner sensitive to long held beliefs.

The co-deployment team should take special care to create and promote positive relationships with any powerful tertiary stakeholders whose consent is required for successful deployment. Their positive influence can be beneficial in generating acceptance from other tertiary stakeholders.

Contextual Deployment Plan. Whilst preparing the deployment, the co-deployment team develops a full plan for deployment in the domain context. Scenarios created in the earlier stages help the team focus on how users and stakeholders can be prepared for and supported in deployment. The plan should be informed by:

- awareness of the context created when working with the domain community
- analysis of the ‘as-is’ skills of the users compared with the skills required to use the new tools
- physical deployment of tools and equipment, including, for instance, the mobile device itself; application and tool set up; and in a mobile context, provision of communications access including how it will be paid for.

For example: the users of the new tools may need training on how to use the physical device, the native tools available, and the applications provided by the technologists. They may need training on the use of the device in the deployment context, including guidelines for professional behaviour, and how to get the most out of the available tools. Secondary and tertiary stakeholders may need information on how the mobile devices might be used in their context and guidelines to support effective use. The preparation and dissemination of information, demonstrations, and presentations should all be included in the co-deployment plan.

Active Deployment

The conversational interaction started in co-design and extended to the deployment community, continues throughout deployment.

Technology and Skills Helpdesk. Primary users are supported with: personal contact with a project support person, a technology helpdesk, drop-in technology ‘surgeries’, a web site with browsable and downloadable content, and on-going skills development with a support person.

Workshops and Focus Groups. Secondary and tertiary users are supported with: a web site, personal contact with a project support person, workshop sessions on deployment in practice, and guidance on supporting users in the deployment context. The successes, concerns, and issues are fed back to the technologists and management team through evaluations, interviews, and focus groups.

Conclusion

We have developed an agile co-design methodology that includes end-users (domain experts) in the design team, in order to help create m-learning applications that innovate practise in a particular domain (as well as using innovative technology).

Our methodology brings together techniques found in HCI (personas, scenarios, and storyboarding), agile software techniques (iterative development and incremental delivery) and lightweight software engineering practice (use cases, simple ontological modelling, and activity diagrams). The methodology is an agile approach that gives us a lightweight method of capturing and recording requirements and feeding these though the design cycle in such a way that they are integral (as opposed to tangential) to the software development process.

Through a case study of creating an m-learning tool for nurses on placement we have shown how the co-design methodology can be effective in identifying subtle requirements, and can result in appropriate application development. The case study also shows that there is a need to take the principles of co-design through to co-deployment, in order to identify and manage the challenges of introducing new technology into an environment with its own particular values and established ways of working.

The co-deployment methodology we have developed in response includes key stakeholders in the deployment team in order to raise awareness of the deployment, and enable the team to identify and address concerns that arise.

We believe that using the co-design methodology can drive the creation of more sophisticated tools that don't just replicate existing practice in a digital form, while co-deployment can help support and foster innovative new practise in the domain based around these tools. Our experiences demonstrate how agile methods throughout the design and deployment can help ease the process of creating new m-learning tools and allow a flexible and adaptable approach that is more sensitive to the eventual end-users.

Acknowledgements

The authors would like to thank the extended mPLAT team, in particular our colleagues Hugh Davis and Mary Gobbi at Southampton, and Balbir Barn at Thames Valley. The mPLAT project is funded by JISC.

References

Abbas, N., Gravell, A. and Wills, G. (2008) Historical Roots of Agile Methods: Where did "Agile Thinking" Come from? In: *Agile processes and eXtreme programming in Software Engineering*, 10-14 June 2008, Limerick, Ireland.

Abowd GD, Atkeson CG, Hong J, Long S, Kooper R, and Pinkerton M (1997). Cyberguide: a mobile context-aware tour guide. Wireless Networks, 3(5), 421–433.

Brown, S. A., Chervany, N. L., and Reinicke, B. A. (2007). What matters when introducing new information technology. *Commun. ACM* 50, 9 (Sep. 2007), 91-96.

Cooper, A. and R. Reimann (2003). About Face 2.0: The Essentials of Interaction Design, John Wiley & Sons.

Collins T.G., (2005) English Class on the Air: Mobile Language Learning with Cell Phones, Fifth IEEE International Conference on Advanced Learning Technologies (ICALT'05), pp. 402-403.

de Crom N., The "ME"-Learning Experience: PDA Technology and eLearning in Ecotourism at TUT, Fourth World Conference on Mobile Learning (mLearn), Oct 2005

Darling, L. (1984) What do Nurses Want in a Mentor? *Journal of Nursing Administration* 14 (10), October 1994: 42 -44

Dix, A., Finlay, J. Abowd, G. and Beale, R. (1993) *Human-Computer Interaction*, Prentice-Hall

Duffy, A (2004) Failing students - a qualitative study of factors that influence the decisions regarding assessment of students, article for Nursing and Midwifery Council, http://www.nmc-uk.org/aArticle.aspx?ArticleID=1675 (last accessed Dec, 2007)

Flynn, M. Pendlebury, D. Jones, C. Eldridge, M. and Lamming, M. (2000). The Satchel System Architecture: Mobile Access to Documents and Services. *ACM Journal of Mobile Networks and Applications*, 5 (4), 243-258.

Gordijn, J. and Akkermans, J. M. (2001) A Conceptual Value Modeling Approach for e-Business Development. In *Proceedings of the First International Conference on Knowledge Capture, Workshop Knowledge in e-Business (K-CAP 2001)*, pp. 29-36 http://www.cs.vu.nl/~gordijn/er2001.pdf

Grudin, J. and Pruitt, J. (2002) Personas, Participatory Design and Product Development: An Infrastructure for Engagement. *Proceedings Participatory Design Conference (PDC*) 2002, 144-161.

Gray, M. and Smith, L. (2000) The qualities of an effective mentor from the student nurse's perspective: findings from a longitudinal qualitative study. *Journal of Advanced Nursing* (32) 6:1542-1549

Highsmith J., Beck k., Cockburn A. and Jeffries R. (2001). "Agile Manifesto" from www.agilemanifesto.org (last accessed July 2008)

Kling, R. (1996) *Computerization and Controversy: Value Conflicts and Social Choices*, Academic Press, San Diego, CA.

Kravcik M., Kaibel, A., Specht, M., and Terrenghi, L. (2004). Mobile Collector for Field Trips. *Educational Technology & Society*, 7 (2), 25-33.

Larman, C. (2004). *Agile and Iterative Development: A manager's guide*. Pearson Education.

Lehman, M. M., (1980) Programs, Life Cycles and Laws of Program Evolution, Proc. IEEE, 68 pp 1060-1076.

Massimi, M., Baecker, R. M., and Wu, M. (2007). Using participatory activities with seniors to critique, build, and evaluate mobile phones. In *Proceedings of the 9th international ACM SIGACCESS Conference on Computers and Accessibility* (Tempe, Arizona, USA, October 15 - 17, 2007). Assets '07. ACM, New York, NY, 155-162.

Parsons D, Ryu H, and Cranshaw M, (2006), A Study of Design Requirements for Mobile Learning Environments, ,In *Proceedings of the Sixth IEEE International Conference on Advanced Learning Technologies* (ICALT'06) pp. 96-100

Prensky, M. (2001). Digital natives, digital immigrants. *On the Horizon*, 9(5), 1–2.

Sharples M., Jeffery N., du Boulay J. B. H., Teather D., Teather B. and du Boulay G. H. (2002) Socio-cognitive engineering: A methodology for the design of human-centred technology, *European Journal of Operational Research* Volume 136, Issue 2, 16 January 2002, Pages 310-323, http://dx.doi.org/10.1016/S0377-2217(01)00118-7

Sloan HL, and Delahoussaye CP (2003). Clinical application of the Omaha system with the Nightingale Tracker: a community health nursing student home visit program. *Nurse Educ*. 2003 Jan-Feb; 28(1):15-7.

Svanaes, D. and Seland, G. (2004). Putting the users center stage: role playing and low-fi prototyping enable end users to design mobile systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vienna, Austria, April 24 - 29, 2004). CHI '04. ACM, New York, NY, 479-486.

Weal, M. J., Cruickshank, D. G., Michaelides, D. T., Millard, D. E., De Roure, D. C., Halloran, J., Hornecker, E. and Fitzpatrick, G. (2006) A persistent infrastructure for augmented field trips. In *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications, Ed-Media 06* (in press), Orlando, Florida, USA.

Wills G, Woukeu A, Carr L, and Hall H. (2003) The need for semantic in web design. Semantic Web workshop, Fourteenth Conference on Hypertext and Hypermedia Nottingham, UK

Yusop, N., Lowe, D., and Zowghi, D. (2005). Impacts of Web Systems on their Domain. *Journal of Web Engineering, 4*(4).