Teaching information communication technology and its development to maritime education professionals

Johan Bolmsten
World Maritime University
Malmö, Sweden

Yvonne Dittrich
IT University of Copenhagen
København, Denmark

Takeshi Nakazawa
World Maritime University
Malmö, Sweden

Abstract

What Information Communication Technology to teach to the Maritime Education and Training community and *how* to teach it?

This paper reports on the development and implementation of a pilot course targeting these issues at World Maritime University in Malmö, Sweden. The course is based on four foundational cornerstones: a.) a problem- and project-oriented pedagogical approach where learning takes place in interaction with a concrete and familiar issue; b.) a light-weight development approach which provides enough guidance for the project group work without getting in the way of an ambition to achieve visible results; c.) a technical development platform that allows for rapid application development without requiring programming skills; d.) the recognition that useful applications require participation of the future users.

In the light of the increasing importance of technology to support both the maritime industry and maritime educational institutes, the main contribution of the course is to academically and practically improve the students relative relation to Information Communication Technologies - may it be as domain expert stakeholders, project managers, or technical developers.

Introduction

Increasingly, technology is an important component for the maritime industry and maritime educational institutes, which presents both new

opportunities and challenges. The growing realisation of the importance of Information Communication Technologies (ICT) in particular have, previously, been reported in the general assembly of IAMU by, for example, Hersham (2003) and Constantinescu (2003).

In light of recent years' rapid advances of ICT, an interest has been expressed to academically and practically explore the application of such technologies specifically by the students of the Maritime Education and Training (MET) program at World Maritime University (WMU) in Malmö, Sweden. The design of a viable course concept, to this end, can be recognised as a challenging endeavour. The course should accommodate students who: have varying but often limited technical experience; come from different organisational positions and roles; and come from different cultures that might have implications for how technical systems are developed and organisationally related. Based on the experiences gained from the implementation of this course within the MET program at WMU, this paper contributes to a discussion of *what* ICT technology to teach and *how* to teach it to the MET community.

In this paper, a concept for a one-week course unit plus homework and experiences of its implementation is presented. The course is part of the third semester specialisation courses within a 17 month MET MSc program. The learning goals of the course unit are twofold: to develop a basic understanding of web-based software products appropriate to support the MET work area, and to achieve a fundamental understanding of how to work with issues of implementation and design of ICT in an organisational context. This way the curriculum improves the students' relation to technology in their daily work, be it as a developer, user stakeholder, or project manager.

The course is based on four foundational cornerstones: a.) a problem and project oriented pedagogical approach where learning takes place in interaction with a concrete and familiar issue; b.) a light-weight development approach which provides enough guidance for the project group work without getting in the way of an ambition to achieve visible results; c.) a technical development platform that allows for rapid application development without requiring programming skills; d.) the recognition that useful applications require participation of the future users.

The outline of the paper has the following disposition: the next section presents the research methods used; section 3 details the above mentioned cornerstones; section 4 reports on the implementation of the course based on the results of the four student project groups; section 5 evaluates the course; and section 6 discusses the course outcome in form of lessons learned.

Method

The concept of the course was developed by the three authors. Bolmsten, together with a fellow colleague at the university, was in charge of implementing the course unit discussed within the premises of this paper.

During the course, an action-based research approach was applied. As an embedded researcher, Bolmsten actively engaged in the unfolding events around him in a process of participatory observation and reflection to understand and account for the targeted activities of investigation. A theoretical ground for the research approach can be found in the principles of "Co-operative Method Development" (CMD) developed by Dittrich et al. (2008). CMD provides a framework for software engineering research combining qualitative empirical research and action research. In this case, it is applied on design, implementation and observation of a course concept rather than software engineering methods.

As an additional input to the research process, the description in section 3 is based on the participating students' own feedback. The account given should not be interpreted as "unscientific storytelling" (Dittrich and Lindberg, 2004), but rather as a way to visualise the area of investigation in order to make it possible for the reader to better assess the work conducted and also argue the conclusions. The comments from the students were given in written form as an input to a focus group taking place after the course that continued to discuss and develop some of the course themes. Approval was given from the respective students when their statements were explicitly cited.

The contents of the course

The goal of the course is for the students to acquire a relevant understanding of the development of a technical tool and at the same time get insights into issues relating to the integration of software in an organisational setting. Instead of presenting a predominately pre-defined body of information in a traditional rostrum teaching setting, both academic and practical active explorations of issues in regard to the areas of inquiry are promoted. In the following, their implementation and how they support each other are explained.

3.1 Participatory design of organisational IT support

Participatory Design (PD) has been developed mainly by Scandinavian computer scientists to support the development of useful and usable software (Floyd et al, 1989). According to our interpretation, PD provides a set of tools, methods and techniques that allow domain experts and software engineers to cooperate around the development of IT support.

Due to the development and deployment of configurable half-products, e.g. contents management systems and ERP systems, the involvement of users in the design, development and maintenance of organisational IT support has changed character. Users today do not only participate in design phases of tailor-made programs, they also select, configure, and maintain configurations. In the article 'PD in the wild', Dittrich et al. (2002) discusses the appearance of a new role: local designers, domain experts that take the lead in the (participatory) design,

procurement, development and configuration of an organisations infrastructure. Those local designers provide the role model to select the learning goals for the designed course. This development is recognised by software engineering research. End-User Development and End User Software Engineering are recognised research topics. (Volker, 2006)

3.2 Project centred teaching using applications proposed by the students

Constructivist learning methods, based on project or -problem based teaching, are state of the art in software engineering (see for example Ohlsson and Johansson, 1995). The purpose is to put students in a realistic situation and support them with a coaching-like supervision to acquire and apply the necessary methods and skills and to reflect on the process and the outcome. It has been adapted to interdisciplinary projects targeting use-oriented design and development of software (Dittrich 1999). These experiences encouraged us to try a similar approach to teach maritime and MET professionals to design, configure and participate in ICT development projects.

An important feature of project based learning is that students are engaged in the projects. To assure this, the students are asked to suggest project ideas of relevance to, for example, the activities of their home organisations. Before the course starts, the suggested ideas are consolidated and an appropriate project scope is negotiated for the selected projects. Working with a project that has tangible stakes, increases the engagement and commitment to the course learning objectives.

From the start, the students are divided into project groups and assigned a project. Throughout the majority of the course the students work towards accomplishing the goals of their projects. To enable a project-based work configuration of the course, technical, methodological, and practical arrangements are put in place. The technical and methodological arrangements are described below. The challenge here was to cut down the project period to one week and still keep the goal of developing a prototype for a realistic application. This influenced the choice of both the implementation platform and the development method.

High level (4GL) technical development environment

The average student attending the course can only be expected to have limited technical background knowledge and no direct prior experience of the technical tool used during the course. This fact puts demands on the choice of the technical platform.

The course uses a fourth generation web-based platform called DotNetNuke that does not necessarily require any traditional programming experience to produce an end product outcome. A complete technical product can be produced through administration and configuration of an existing framework and existing software modules. Today, there are a wide variety of similar technical alternatives available both in the open source community and in the commercial market.

Practically, the framework acts as a host environment containing aspects such as menu and -security management. The framework also becomes the canvas on which different pieces of functionality are placed, such as document management, announcements, forums etc. The different components can both be built from scratch or pre-manufactured and ready to apply after configuration. In addition to providing a technical platform during the course, a collection of components are selected and made available to the students based on the character of their different projects.

3.4 A light-weight software development method

The course uses a light-weight software development methodology called extreme programming (XP) (Beck, 2000) which also works as a template for teaching pedagogy. XP is a project-centred approach. The focus is on empowering a local project group to design a software solution with high requirements on usefulness and usability. The development work is structured by a number of guiding development practices that both individually and in conjunction define best practices for the different dimensions of a software design process. Schematically, they can be described as follows (practices are in italics):

An *on-site customer* opens up for a PD approach during development. A project-centred style of work is upheld by *collective ownership* of the solution designed by all involved project members, and that all development is done through pair *programming*.

A project is guided by an overarching *metaphor* that is broken down into concrete requirements through a *planning game*, where individual pieces' functionality are specified from the customers own descriptions on story cards. The importance of realistic planning and expectations is emphasised by the practice of a *40-hour week*.

An evolutionary mode of development is promoted through developing functionality in *small releases* with, at any given time, the most *simple design* possible that is *continuously integrated* into an operational product. Other important practices that steer the technical development are following *coding standards* and performing continuous *testing* and *refactoring* of developed functionality.

A fundamental difference between XP and a traditional software engineering style of software development is that XP does not rigidly prescribe certain procedures or a particular implementation style. Instead, a method should only be used to the extent that it fulfils a relevant purpose within the local project.

In summary, the technical high-level development environment resembles a complex but tailored software product. Learning to handle its possibilities and constraints enable maritime professionals and MET professionals to cope with similar complex customisation and configuration tools. The XP approach to software development provides a frame that guides and encourages cooperation between domain experts and IT professionals. In this context, the PD principles put a strong focus on the capabilities of domain experts as local designers.

The implementation of the course

As a foundation for evaluation, this section describes the work process of the different projects and the resulting ICT products. An individual account is given for each project where some of the central characteristics are highlighted. In this way the reader is able to get a more multi-faced insight into the effects of the course design.

The particular instance of the course described in this paper was set during one week. For five days studies were scheduled between 09.00 and 14.00. To complete their projects the students were also expected to perform after class work.

The course was located in a purposefully arranged classroom where the furniture was re-arranged to form four personal project work areas; one for each group. Every project work area was equipped with two computer work stations (to support the pair programming practice). The classroom was open both during and after class hours.

During the two first days of the course, two theoretical sessions were provided and complemented with practical exercises to achieve a basic familiarity with the different components to be used. This also enabled the students to academically question their experiences. The three remaining days were devoted almost exclusively to project work. In the end of the course, each project group presented their final product as well as their work process to all the members of the class.

	Group 1	Group 2	Group 3	Group 4
Project	Student and staff intranet	Community portal	Staff intranet	Sea Farers Portal

	Group 1	Group 2	Group 3	Group 4
Metaphor (developed by the groups)	Keeping You Afloat	maintain links and develop the maritime professional network striving to achieve *one team one goal*	Knowledge creation and utilisation promoting dynamic system of creative routines	Waypoint To Seafarers' Welfare
No. of members	4	4	4	4

Figure 1. Overall project charter

4.1 Student and staff intranet: group 1

The majority of students came from different maritime education training institutes and an interest therefore existed to design a platform similar to the one implemented at WMU. It was decided that group 1 was going to focus primarily on student and staff communication.

4.1.1 The product

The resulting portal presented an overall proposal of how a working student and staff intranet portal might be organised based on the students own experiences from their home organisations. It included relevant pieces of functionality for this purpose including text/picture presentations of programs and subjects; general information and overall presentation of the different departments of the university as library and health care; document management to share files; and events modules to publically calendar the university's activities.

4.1.2 The group process

In the beginning of the project work, it was noticeable that members of the student and staff intranet group had little prior experience with technology. Unexpectedly to the group members themselves, it therefore turned out that *pair programming* became one of the most valuable development practices to the project (Overall, pair programming is arguably the most debated development practice. Its value is emphasised in XP, but it often departs from what professional system developers feel comfortable with.). As a student expressed:

"at first, it seemed to me as unhelpful, where we probably could waste more time than be productive, but in the end, it resulted to be highly beneficial for the project...we experienced how the mistakes were easily detected by the observer, which kept us on track". Because familiarity with technology was, in general, low in the project, substantial time was required for the group to make sense of the technical tool itself. This naturally implicated the extent to which the students worked with the other development practices. Accordingly, in addition to *pair programming* the other practice that the students mainly referred to was the *metaphor* for overall steering of the project. Another student stated regarding the metaphor: "it was the one we most often came back to...[it reminded] all those involved about how the project is supposed to satisfy the need of the customer". Arguably, the students did recognise that the practices acted as separate yet connected parts of a whole, but as a consequence of the focus on the technical issues of the project, it was stated that "it is not easy too see a crystal clear difference between them [the practices]".

4.2 Community portal: group 2

This portal took the form of a virtual seafarers lounge where current and graduated students can meet and where outside professional are also invited. Portals expanding the borders of educational activities and facilitating networking is a hot topic at WMU and at other related universities.

4.2.1 The product

The end result had a strong focus on communicational aspects such as forums and chat modules. The community portal, possibly left the widest scope of interpretation with a broad array of functionality that could be considered for inclusion. This was reflected in the metaphor chosen by the project group: "maintain links and develop the maritime professional network striving to achieve *one team one goal*". The students also considered how to work with different layers of security to diversify the access to different parts of the portal depending on which user group a user belonged

4.2.2 The group process

To cope with the constraints of the project such as the limited time available, the students actively worked with both the planning game and with the development practice of simple design to determine the scope as well as to prioritise the functionality. These practices also made the students able to understand the purpose of an evolving design, where it is possible to expand a piece of functionality in iterative cycles. One of the project members elaborated on these concepts as follows: "As we started, we thought of a perfect webpage

with beautiful design and cool functions, informative and attractive. Then we faced a lot of problems like time constraints. Conversely at least, we have launched our project on time with all necessary functions even we all know that it needs to be improved...We had to try to think like a manager that what is the minimum functional or information that need to be included in the first releases which were supposed to be launched in two weeks time...How much of a problem has to be solved in order to achieve value in production. Using story cards and simple design helped us a lot to finish out project properly".

4.3 Staff intranet: group 3

The staff intranet was a project already under consideration at one of the student's home university. Comparably, it can be argued that the staff intranet group had the most tangible project. It appeared as a project that all the project group members personally could relate to within their professional contexts.

4.3.1 The product

In terms of functionality, the project shares many similar features with the student and staff intranet but with a primary focus on staff related issues. This, for example, includes saving and sharing hand outs to be distributed to the students in class. The staff intranet became the most technically-oriented project which among others resulted in that the project members worked more than the other project groups with customisation of the implemented functionality.

4.3.2 The group process

Unlike the community portal project, an explicit customer stakeholder existed who guided the project towards a manageable set of prioritised features during the work of the course week. The student that the project idea originated from the following evaluation: "when the project started, the customer did not have a clear picture of the final product. Due to this reason, new stories were developed continuously during the whole project... The customer was sitting with the team so he could update the plan".

As a consequence of the clear project objectives provided, the project work assumed a pragmatic technical oriented development focus where the group members to a large extent concentrated on the implementation of functionality. This became reflected in the extent to which the group used the design practices. As another project member stated concerning the limited use of story cards for the planning: "we may have 6-10 such cards, with one sentence each". As the group did not perceive any purpose to work with the design practices as a whole, consequently they did not get any deeper insights in their broad application. One of the group members recognised that the practices "counterbalances and

reinforce each other" but "picking and choosing which to use and which to discard can also be tricky" and "the difficulty is not the result of technical challenges but of social challenges".

The group's technical orientation also enabled them to "challenge" the functionality to the point that "somebody initiated to do some changes that he thought it is would be useful to our page. Unfortunately the whole project was gone after that".

4.4 Seafarers portal: group 4

In the customer's own words "The project had a real business representative who desired the software and who would be the administrator." Of all the projects, the Jamaican seafarers' portal had the business stakeholder or student customer with the most outspoken aims and defined vision of the project outcome. The project related to an identified need in regard to national sea administration and primarily concerned the fact that seafarers' on different levels of a nation's ship fleet can benefit from a being able to share information.

4.4.1 The product

As the community portal, the functionality of this portal came to focus on communicational aspects such as forums. Information published can relate to multiple topics, for example incident reporting and sharing of best practices. In addition, central administrative agencies have a need to be able to distribute information which can concern new regulations or policy changes. A web-based portal would facilitate such communication as the parties are often at sea and therefore geographically separated.

4.4.2 The group process

When the project members started to work with the technical tool and the development practices, different opinions surfaced on what the project was going to entail. These were not easily reconciled. As described by a project member: "In the beginning; the planning game process for our group does not seem to run very smoothly. Even though there are positive discussions among the group members, it is still not easy to get an agreement among the members." Due to this circumstance, the project members were referred to actively explore how the development practices could help guide or mediate a satisfactory project process. After a suggestion by the lectures, it was also decided to balance the different interests in the project by assigning/enforcing the stakeholder roles of 'programmer' and 'coach' with clear mandates in addition to the already existent customer. As a result of these efforts, it was stated that: "Later on as the members begin to understand their specific role and acknowledged the role of

other group members -- especially the customer -- things start to work smoothly." The customer appeared satisfied with this process and by the time a metaphor was selected she stated that "The initial resistance waned and all team members accepted their roles because they had a part to play in developing the metaphor, so in essence they had to commit to what they invented. It became not just the customer's project, but a team effort with different roles."

Due to the reflection process regarding their course of work that the group had to undertake, they managed to get a comparably in-depth application of how the development practices individually worked but also holistically interacted. This can be exemplified by a mutual report submission from the group: "The story cards gave direction in the *planning game* and in mapping out the scope of the project for *small releases*. They gave life to the *metaphor*. Having story cards ensured a systematic and *simple approach* was taken in the development of the system allowing for continuous *testing* and sometimes immediate changes, corrections etc. The story cards kept the focus and allowed any member to continue with, according to the practice of *collective ownership*." The group was also able to critically point out unresolved issues in their experience with the design practices. Their above average understanding of the work process was also reflected in their combined grades for the course.

Evaluation

As stated in the introduction, the course unit had two learning goals: to develop a basic understanding of web-based software products appropriate to support the MET work area, and to achieve a fundamental understanding of how to work with issues of implementation and design of ICT in an organisational context. This section evaluates the teaching approach and whether the learning goals were achieved.

5.1 Project-based teaching supports integrated learning of technical and process related knowledge

The Seafarer Portal project provides a good example of how integration between technical and process-related learning works out: During the initial stages of the group work a conflict arose concerning the direction of the project. This challenged the group members to discover and reflect over how the development practices could be utilised to improve the development of their particular project. In the end, the seafarers' portal group managed to achieve a successful outcome in terms of creating a balance between understanding the technical environment and the applicability of the development practices.

In accordance with a project-based perspective on learning, it is possible to deduct from the work process of the different projects that facing problems do

not mean that one does not learn. As illustrated, problems encountered in the project work are, in fact, possibilities to learn. In this process, the teachers act as facilitators or coaches whereas the conflict remains with the students. Instead of adopting a passive and receptive role, the groups are supported in experimenting with the methods and tools provided. To this end, the learning environment should be designed to support and challenge the learner's thinking.

5.2 Basic understanding of web based software products

Though the students had various understandings of technical software support when entering the projects, all of them gained basic knowledge regarding modern ICT and how to use it to support organisational tasks. Only a few of the course participants had prior experience working with software development per se. Given these preconditions, the technical sophistication of the resulting products was apparent in the final presentations. One group, even, clearly reached the limitations of the DotNetNuke platform, when trying a configuration that resulted in a major breakdown. Such experience (for software engineers in general) allows reflection on the implementation platform as an enabling tool as well as a limitation.

All the project groups showed a good capability to translate their ideas into technical pieces of functionality. In other words, they managed to conceptualise how a certain task, for example a distribution of hand-outs for a course, could be technically supported by a particular software module, i.e., a document management module. This also includes developing an understanding of the role of supporting functionality as management of menus, and security settings.

5.3 Implementation and design of ICT in an organisational context

The course assumes from a diversified view of who is the designers of system development projects. Even though it is not a prerequisite that the students have any experience with technology, they are - confidently- put in the driving seat of the design of software. Especially, the students' visions of how their solutions could continue to evolve in order to show their ability to understand the relationship between software support and organisational activities (also see section 6).

The problems encountered during the design process mirror real-world problems: For example, the members of the seafarers group encountered differences of opinions about the direction of the project. Stakeholders who, due to their different perspectives of an application domain, have different opinions

of what constitutes a satisfactory solution can be recognised as a fundamental issue in a live development setting. The community portal group faced another common challenge regarding balancing users' exceptions on functionality with what is technically possible to implement given resource constraints such as time and knowledge. This made the participants actively work with both prioritisation and the scope of the chosen functionality.

Discussion of the course rationale and design

This was the first time that the course was given. Although the founding rationale, as described in section 3 is based on established practices of teaching software development, the course concept in a MET setting is, to the authors' knowledge, new. The course, therefore, became a pilot platform to investigate the teaching and developing approach as such and the applicability of the individual design practices.

6.1 Teaching just enough software engineering

The twelve XP practices applied during the course allowed insight into basic software engineering issues: roles are a fundamental way to assign responsibilities in software engineering; design artefacts – notations, documents, and their purpose in the development process – are central to most software engineering methods; planning and prioritisation are central tasks of project management; pair programming and nightly builds add a quality assurance dimension, and so on. To support the notion of the local designer introduced in section 3.1, the practices gave the project group members a light-weight frame for their group work, providing them with just enough structure, without introducing unnecessary overhead.

Whether the students can apply the entire framework of development practices in their home organisation remains to be seen. As one of the students noted "This subject is closely related to my work as it is heavily uses websites as a main tool... with 12 XP practices, that knowledge can be very helpful in my similar works in the future even though not all of them can be applied..." The end part of the last remark was expressed by many of the students. The complete application of the development practices promote a certain way of organising that might not be in line with the prevailing way of working in the students' home organisations.

As an improvement, it is proposed to reflect more explicitly on the XP framework as one way to address the challenge of software development and emphasise the above mentioned basic issues addressed by the specific practices. The development practices can, however, provide value individually as well as in combination. They may, for example, be applied to both improve an existing

work style one step beyond the current level of mastery or as springboard for more fundamental re-invention.

6.2 The high-level (4GL) technical development platform: enabling tool and constraint at the same time

The teaching style applied during the course would not have been possible without the technical platform chosen. If traditional programming were required to implement the students' portal projects, then the outline of the course would have to be fundamentally re-arranged.

However, even though the high-level nature of the DotNetNuke platform enabled the students to conduct the technical parts of the development, it also poses constraints. In the middle of working with the configuration of a piece of functionality, the staff intranet group solution broke down. An intervention was required by the teachers to restore a backup in order to get the portal up and running again. Though the students mastered the majority of the development themselves, the teachers actively facilitated the development process by acting as resource persons with technical expertise. Adequate support is necessary, and the role of IT professionals should not be dismissed inside or outside the classroom context.

6.3 How to complement the project work to support more equal learning

As mentioned in section 5.1, the project-based approach was a success as it engaged the MET students and thus facilitated learning results that otherwise would have been difficult to achieve. To this end, an illustration is provided of a successful application of a constructivist teaching method – a teaching style which is not only state of the art in software engineering (as described in 3.2), but in line with the objectives of the STCW95 convention of growing importance also in the MET community (Asyali et al 2003; Fedila, 2007; Horck, 2007).

By working with authentic tasks, relating to the students' own professional domains, it was possible to increase the students' motivation and enhance their confidence to engage in the work process. However, as each project developed in a specific way, the learning results also differed. Let us compare the Seafarers Portal project with the Staff Intranet project: As reported in section 4, the specific configuration of interests in the Seafarer Portal project resulted in a thorough discussion on roles and responsibilities. In the Staff Intranet project, the common experiences resulted in a much more homogeneous understanding

of the development practices. Thus the process was smoother but resulted in less learning regarding how to handle stakeholder conflicts in software engineering.

There are different ways to address this problem: One could introduce teacher-guided, class-wide meetings on a daily basis where the different projects, their challenges, and how to address them with the help of the development practices are discussed. This would give the students an opportunity to discover practices outside the immediate horizon of their projects and thus get a more holistic appreciation of the development practices. Another way could be to introduce scenarios that highlight everyday challenges of software development. These measures would work as a complement to the post reflection provided through the mutual presentations of the product outcomes and the development processes in the end of the project period.

6.4 How does culture affect the applicability of software development methods and practices?

As mentioned in section 5.3, the students of all projects managed to relate the design of technology to organisational needs. According to our evaluation, all students will be able to take part in a software design and development project as domain experts and user representatives and in that way work towards a match between the software under development and the needs of users and organisation. However, during the discussions with the students, the applicability of the PD principles and the XP development practices – that implements a very egalitarian way of organising software development – in non-western or non-Scandinavian cultures was questioned: "[T]he most difficulties of implement the XP practices are the attitude and the culture of the organisation." "The culture organisation is hierarchical with division of authority and responsibilities. The information system which flows from the top management to the low level currently do not apply any software with the use of XP practices."

To discuss these cultural issues in depth would go beyond the scope of this paper. (See e.g. Horck 2006) In the context of this course, local designers – and with this a PD perspective – (Dittrich et al. 2002) are primarily used as a role model motivating the learning goals and the XP framework as a light-weight process support for the student projects. However, the interaction between software engineering methods and cultural diversities -- respectively the need to adjust the methods and approaches to different cultural backgrounds -- was and will continue to be subject to reflection during the course as well.

Concluding remarks and future work

As a response to the increasing opportunities that comes with ICT to support MET, this paper has reported on the development and implementation of a course that gave the students both academic and hands-on appreciation of such technologies. During an intensive course week, the students of a MET program worked with projects relating to their own areas of interest to investigate the usability of a technical tool and to understand issues relating to usefulness in an organisational setting. The central lessons learned are:

- Given the students' limited pre-knowledge about technical tools, it was apparent how competent the technical product outcomes became. The importance of adequate support was, however, also evident. The role of the IT professional is not dismissed.
- The students' capabilities as designers were emphasised. The course presented an all-encompassing method for software development with high-requirements of usability and usefulness. Although due to organisational and cultural constraints it might turn out difficult to transform the entire model into practice outside the classroom, it was possible for the students to actively reflect over how they can improve their role as stakeholders in their future professional careers.
- The course also gives a positive account of the experiences of using a project-based approach to learning. It is believed that this increased the students' motivation to take ownership of their own learning process.

An interesting future topic for research is to investigate what use the students actually had of learning how to apply ICT technology. In regard to the specific projects worked with during the course, so far two of the members in the student and staff intranet group have expressed an interest to continue to individually develop their application for their home organisations. The entire community portal group will continue the development together. It has been decided that it is appropriate to host the portal as a separate entity in connection to WMU's own community portal. In the case of the staff intranet and the Seafarers portal, they are under review of the home organisations of the students that suggested them.

Acknowledgements

We thank the students for their engagement in the course and for giving input to this paper.

References

[1] Asyali, E, Zorba, Y, Nas, S 2003, 'Adaptation of Problem–Based Learning Method to Requirements of STCW Convention', *IAMU 4th IAMU General Assembly*

[2] Beck, Kent, 2000, *Extreme Programming explained : embrace change*, Addison-Wesley, cop

[3] Constantinescu, E 2003: "TIME Model or How to Integrate New Technology on Maritime Education and Training", *IAMU 4th IAMU General Assembly*

[4] Dittrich, Y 1999, 'Shopping Web-shopping Web-shop Web-selling Selling. Teaching Software Development together with Work Practice Studies.' *Proceedings of the IRIS 22*, 7-10 August 1999 Keuruu, Finland.

[5] Dittrich, Y, Eriksén, S, & Hansson, C, 2002: 'Participatory Design in the Wild; Evolving Practices for Design in Use.' *Submitted to the PDC*, Malmö, Sweden.

[6] Dittrich, Y & Lindberg, O 2004, 'How Use Oriented Development Can Take Place', *Information and Software Technology*, vol. 46, no. 9, pp. 603-617.

[7] Dittrich, Y, Rönkkö K, Eriksson, J, Hansson, C, & Lindberg O 2008, 'Co-operative Method Development Combining qualitative empirical research with process improvement', *Empirical Software Engineering JournalI,* vol. 13, no. 3, pp. 231-260.

[8] Fedila, Mokrane , 2007, *Appropriateness of Problem Based Learning in Maritime Education and Training*, dissertation Master of Science, World Maritime University, Malmö, Sweden.

[9] Floyd, C, Mehl, W-M, Reisin, F-M, Schmidt, G, & Wolf, G 1989, 'Out of Scandinavia: Alternative Approaches to Software Design and System Development', *Human-Computer Interaction*, vol. 4, no. 4, pp. 253-350.

[10] Hesham, H 2003, 'Computer Based Training: A Global Survey of Current Developments and its Application to Maritime Education and Training', *IAMU 4th IAMU General Assembly.*

[11] Horck, Jan, 2006, *A Mixed Crew Complement. Licentiate dissertation in Education. Malmö Högskola*, Sweden

[12] Ohlsson, L & Johansson, C, 'A Practice Driven Approach to Software Engineering Education', *IEEE Transactions on Education*, vol 38, no. 5, pp. 291-295.

[13] Volker, W, Lieberman, H, & Paternó, F (eds.) 2006, *End User Development: Empowering people to flexibly employ advanced information and communication technology*, Springer.