GrainPlan; The Virtual Grain Store Manager

Julie Shields a, Caroline Parker b

a Glasgow Caledonian University, Glasgow, Scotland, Julie.Shields@gcal.ac.uk
b Glasgow Caledonian University, Glasgow, Scotland, Caroline.Parker@gcal.ac.uk

Abstract

GrainPlan is an electronic grain store manager which is designed to help farmers manage storage data relating to grain quality and to improve their ability to respond to problems. This paper describes the process of the GrainPlan Project, the arrival at the final product and the reasons for its success, providing information about the technical and business processes involved with the implementation of this project. The paper will take an honest look at the problems as well as the successes, and how lessons learned can help shape the process for similar projects.

Key words: Grain store management, user-centred software, data modelling, usability, software project management

1 Introduction

Maintaining the quality of grain in store is becoming increasingly important, particularly as our knowledge of the relationship between store contamination by pests and moulds and public health increases (Hicks et al 2000). The GrainPlan project was launched by the UK’s HGCA (Home Grown Cereals Authority) in November 2002 under the Grain Sampling and Analysis Project (GSAP) programme funded by the HM Treasury Invest to Save budget. This programme had generated a great deal of valuable scientific information about the behaviour of grain in storage and in grain sampling (Armitage, 2003). However the new knowledge requires more in the way of monitoring and recording on the part of the grain store manager if the benefits were to be realised. The aim of the GrainPlan project was therefore to provide grain store managers with a tool which would reduce some of the effort they needed to put in, to support the storage and monitoring of grain store quality data, and thereby aid the improvement of UK grain quality.

Version 1 one of this project adopts an inclusive design approach and concentrates on the needs of the user, therefore helping us to produce initial requirements as well as the build of a usable product. The second phase of GrainPlan increased the amount of functionality but maintained the same level of resources to work with. Only the future release of version 2 will determine its success in terms of user acceptance. This paper describes

a) the features and development of the software
b) the user involvement with the project
c) feedback to date from our users.
2 GrainPlan Development

2.1 Project Method

There have been two phases in the GrainPlan project to date. The first phase started in January 2003 and ended in January 2004 with the release of GrainPlan V1. The second phase started in February 2004 and will end with the release of version 2 in June 2005. The software is developed in Delphi™ and currently uses a Microsoft Access™ database. It is capable of running on Microsoft® Windows 98 and upwards. The hardware specification for GrainPlan requires an Intel Pentium II processor or equivalent with a clock speed 300MHz or higher and a minimum memory requirement of 64MB RAM.

Although there are other farm management software packages available on the market GrainPlan is a bespoke piece of software and required a variety of professionals to produce the product. Our team of multidisciplinary professionals included agricultural scientists, computing professionals, a psychologist, a human factors specialist and the user community. Abras (2004) suggests that user-centred design teams generally benefit from including persons from different disciplines, particularly psychologists, sociologists and anthropologists whose job it is to understand users needs and communicate them to the technical developers in the team. The downside of this approach is that members of the team have to learn to communicate effectively and to respect each other’s contributions and expertise. This can be time consuming and hence adds costs to the process.

Using a traditional phased approach combining iteration we were able to accommodate the changes in user requirements and clarify the stakeholders’ understanding of what the system should do (Krol et al, 2003). Although this process was time consuming and the project encountered a few iterations we believe the final build of GP version 1 was intuitive to the users due to their continuous involvement. Locating a simple lifecycle that considers HCI techniques along-side the role of Software Engineering proved difficult. The Usability Engineering Process Model (UEPM) (Granollers et al, 2003) best describes the methodology our project adopted. See fig.1

![Usability Engineering Process Model](image_url)
Version 2 however did not follow the same path as the first release and we were not able to involve the users as much as we had hoped. This stage was led by mainstream software design principles, such as producing a set of full requirements upfront, coding these without coming back to the users or the client until completion. The outcome was the inevitable requests in changes to the requirements and the trade off between robust functionality and usable interfaces. The added functionality and project problems that we encountered resulted in running over time and budget, an account true of many software projects. Although developers would like to know what attributes to incorporate into the code to “reduce required effort for its use”, the presence or absence of predefined attributes cannot assure usability, just as there is no trustworthy way to predict the behaviour of the users with the end product (Bevan, 1999). Due to the approach taken with version 2 it suffered a lack of user involvement and we look forward to receiving the results from the user group.

2.2 Features

User feedback from version 1 suggested that the system would need to support the collection and collation of the vast amounts of data that grain store managers are now being asked to collate. It also suggested that they would like dynamic access to the results of the scientific data generated in a number of detailed studies on grain spoilage i.e. the level of danger represented by a specific temperature and moisture reading and also access to background information e.g. about pest control. The main user interface to the system is a customisable representation of the layout of the user grain store as shown in fig 2. On entering temperature and moisture data the system will flag up the extent of any risk and provide links to sources of information about how to deal with them. The encyclopaedia is designed and built within DESSAC™ (Decision Support System for Arable Crops) browser. DESSAC is designed in a scalable manner to accommodate the addition of other modules and as a result bringing a wide range of support to husbandry decisions (Arable Decision Support, 2005).

Version 2 continues to accommodate the useful benefits mentioned previously but with the added functionality of the recording of quality assurance data and its reproduction in a form that is acceptable to the main farm assurance schemes within the UK. This feature saves the Grain Store Manager additional time, and time was a problem that ranked high on our user feedback list. Additional requests were for ways to reduce the double handling of data that comes from recording in the grain store but having to interact with the program in the farm or store office. Version 2 makes use of a PDA recording tool and simple data entry interface which allows the user to upload data effortlessly when they return to the PC. These features were derived from a series of user training workshops that took place at the HGCA Roadshow during January 2004 through to March 2004.
The main features of GrainPlan are therefore its ability to provide information about loss of grain quality, providing support about grain sampling and throughout the process enrich the user experience with a comprehensive encyclopaedia containing a wealth of information about identification and treatment of grain store problems as shown in fig 3.
2.2 User Involvement and Feedback

It has been argued that low uptake of model-based systems is largely due to the lack of attention paid to users’ needs by system developers. Involving those who will eventually use the software can help avoid the problems that cause a lack of uptake in software from farmers and growers (Parker et al, 1997). The main objective with version 1 was to provide a user friendly product which supported safe grain management resulting in the enthusiastic acceptance of the software by its user group.

As our project followed an iterative approach, during the development of version 1 we included the users at three different stages as suggested by Mayhew (1999).

- The first level evaluation is an iterative conceptual model evaluation, designed to get feedback before any code has been developed.
- The next testing stage should be done after the prototype has been coded to get early feedback about its usability.
- The third testing phase occurs after the interface is ready, and its purpose is to evaluate the final product against the usability goals set at the beginning of development.

The GrainPlan project adopted a user-centred design approach from the outset and experienced three different stages of user involvement as described by Mayhew (op cit.). Initial meetings were held with representative users before any detailed design took place. Once a working prototype was ready it was used to gain additional feedback from farmers, consultants and grain store managers. The prototype then formed the basis of a working system which received another round of testing and feedback by our group of approximately 20 beta testers before being completed and distributed. A final opportunity was available at the HGCA Road Shows which are made available to levy payers. Here we carried out user training and a final evaluation of the usability of the user interfaces. As we were already aware at this stage that GrainPlan was to be phased over two versions any feedback gathered during the final meeting with the users was fed into the development of version 2 which is also befitting with the UEPM shown in fig 1.

The deployment of version 1 proved to be a great success. The simple nature of the user interface and the drag and drop interaction was seen to be both useful and intuitive. Version 1 has been made available to all HGCA levy payers and while it is not possible to estimate the actual amount of users because the CD is widely distributed free of charge, most of the responses received to date have been very positive.

Version 2 is more complex, largely as a result of the additional features requested by the users, and we have yet to find out how this will impact on its usability.

3 Summary and Conclusions

The GrainPlan concept has been very successful and both versions provide the features requested by the users, such as easy data entry, complete storage records, monitoring for safe storage, warnings and actions, a comprehensive encyclopaedia, detailed reports and PDA entry. This is definitely a product with a real niche in the UK market. The user interface to version 1 was also an undoubted success, largely we believe due to its simplicity and the extent of user input into its design and development. Other software products in the area of crop production have experienced similar success and studies provide positive evidence that this is due to the employment of user-centred design (Parker et al, 2001).
The development process with version 2 has not been easy. Whilst the software itself meets the desired features the project was beset by a variety of problems that are true to a lot of software projects using a similar approach. Omitting the need to accommodate changing requirements have always been primary sources of project trouble, leading to late delivery, dissatisfied customers, and frustrated developers. To address these problems, teams who use an iterative approach focus on producing and demonstrating executable software in the first few weeks, which forces a review of requirements and helps to pare them down to essentials. (Kroll et al, 2003). Version 2 has consequently been developed in a less user intensive way and we believe may suffer as a consequence. Only the feedback from the June 2005 release will enable us to comment on this.

5 References


Dessac developer website, (2005,) http://www.dessac.iacr.ac.uk/


