T-76.5158 SEPA diary

Managing a distributed agile project
(Agilefant)

Teemu Kiviniemi
Pekka Latvala

Revision History

Revision 0.10	18.10.2007	Teemu Kiviniemi
First version		
Revision 0.20	21.10.2007	Pekka Latvala
Start of writing user manual		
Revision 1.00	22.10.2007	Teemu Kiviniemi
Version for PP iteration document submission.		
Revision 1.10	13.11.2007	Teemu Kiviniemi
Initial conversion to DocBook format.		
Revision 1.20	14.11.2007	Pekka Latvala
Modified defect tracking, collecting data		
Revision 1.21	22.11.2007	Teemu Kiviniemi
Minor DocBook fixes. Updated: Section 1.1, “The problem” , Section 1.2, “The proposed practice” , Section 2.3, “Before each sprint” , Section 2.5.1, “Collecting data” ,title. Removed unneeded text copied from the course template.		
Revision 1.3	10.11.2007	Teemu Kiviniemi, Pekka Latvala
Added I1 iteration experiences.		

Table of Contents

1. Introduction 1
1.1. The problem 1
1.2. The proposed practice 2
1.3. Collaboration with the Agilefant software project team 2
2. Deployment of the practice 2
2.1. Installation 2
2.2. Training 2
2.3. Before each sprint 3
2.4. During each sprint 3
2.5. After each sprint 5
3. Experience and changes 6
3.1. Project planning iteration (PP) 6
3.2. First implementation iteration (I1) 6
3.3. Second implementation iteration (I2) 9
3.4. Conclusions 9
References 9

1. Introduction

1.1. The problem

We're using a Scrum-like [ScrumAlliance] software development method for this project. Scrum is an agile, iterative development method. The iterations, sprints, have got a fixed duration. After each sprint, the software development team delivers an increment of new code and features that is tested and documented.

All the requirements of a sprint are kept in a sprint backlog. The backlog consists of backlog items, each of which describes a task to be done or a feature to be implemented. Before each sprint a suitable amount of backlog items are planned for implementation and added to the sprint backlog. During a sprint the contents of the backlog are frozen; no new requirements may be added to the backlog.

Optimally everyone in a software development team would work in the same place at the same time. But in this project that won't probably happen. People will work in different places at different times. The distributed nature of this project creates challenges for project management. It is harder to know what everyone has done, is doing or will be doing next. Getting a good overview of the status of each sprint is crucial for project management to be able to manage risks and spot problems before they get out of hand. The end of a sprint is a deadline that can't be missed.

To sum it up, an efficient solution is needed to manage the backlog items and to measure the progress of each sprint.

1.2. The proposed practice

Agilefant [Agilefant] is a backlog management tool for agile development methods. It can be used to manage the backlogs of several products, projects and iterations. For each iteration several goals can be defined, and several backlog items can be linked to each goal. If backlog items are too big, they can be split into smaller pieces that are called tasks.

The tasks and backlog items in Agilefant contain description, priority, the amount of effort left to complete the item and the person who is responsible for that item or task. Backlog items can also be moved between iterations, and for example, from project backlog to iteration backlogs.

For each iteration Agilefant provides an automatically generated burn-down chart that can be used to measure sprint progress. The burn-down chart shows the amount of work left as a function of time. The amount of work left can be compared to a steadily advancing iteration.

We will study using Agilefant as a tool for project management. We want to know how it helps project management, how users feel about using it, and how much effort using it takes. All found problems will be reported and improvement ideas will be given.

1.3. Collaboration with the Agilefant software project team

We'll be working closely with the Roni and Ilkka, a SEPA pair of Team Maranello. [MaranelloSEPA] Team Maranello is the software development team that is working on Agilefant on this course. They will provide us Agilefant training and support, and we will provide them feedback, questions and suggestions.

We are using a shared blog [Blog] to communicate with Team Maranello. The blog will be available at http://agilefant.blogspot.com/ and it will be written in Finnish.

2. Deployment of the practice

2.1. Installation

Agilefant will be installed on a server provided by our project customer. All members of the group, our mentor and our customer will receive a user account to Agilefant.

2.2. Training

Before the first sprint, training material will be given to all group members. The instructions will be written in Finnish and they will be available in the group wiki at http://tiger.soberit.hut.fi/

T-76.4115/2007/10/mediawiki/index.php/Agilefant-opas . In addition to that, we'll try to arrange face-to-face training to group members.

2.3. Before each sprint

During the planning of each sprint, all requirements will be entered to Agilefant as prioritised backlog items with effort estimations. If a single backlog item takes longer than 12 hours to implement, the backlog item is split into smaller tasks.

Each task and backlog item will be assigned to a team member. The person will be responsible for updating the status of the backlog item during the sprint.

Backlog items may be created also before sprint planning. Such backlog items must be polished during sprint planning if needed.

2.4. During each sprint

All group members will use Agilefant from the beginning of the first sprint. During sprints people responsible for each backlog item must update the status and estimation for the amount of work left to complete the item. The effort estimations are updated daily if they have changed.

From the given input data Agilefant provides burn-down charts which are used by the project management to measure sprint progress. If it seems probable that all the selected backlog items can't be completed during the sprint, the scope of the sprint can be reduced to minimize risks.

2.4.1. Defect tracking

Project's defect tracking will be done with Agilefant. Because Agilefant is not specially designed for defect tracking, there has to be strict guidelines for reporting a bug.

Normally all bugs will be fixed as soon as they are found but if they cannot be fixed right away, the bugs must be clearly reported for later fix. In Agilefant each bug will be presented as a backlog item and placed to the backlog of the ongoing sprint.

The information that will be inserted to Agilefant is:

2.4.1.1. Name

The name of a bug item should be BUGXXXYYY, where

- XXX is the ID of the corresponding PR-item in Agilefant backlog
- YYY is the running number starting with 001

It is important to notice that the corresponding PR-item is not always clear. That is why everyone is recommended to use 000 as XXX in cases of bugs which cannot be clearly categorized to any PR-item. It is also important to notice that every XXX has its own number space for running YYY.

2.4.1.2. Original estimate and effort left

The bug item must contain also an initial estimation of effort required to fix the bug and it should be updated if needed.

2.4.1.3. Status

Agilefant statuses are mapped to bug statuses having the meanings presented below:

Bug status	Agilefant item status	Meaning
Detected	Not started	Bug is detected and described, but nothing has been done to fix it

Bug status	Agilefant item status	Meaning
Fix started	Started	Someone has started working on a bug fix
Fixed	Implemented	Fix is implemented and all the original symptoms have disappeared but the needed tests have not been run to verify that the fix does not break anything else
Fix verified	Done	Fix is tested to both fix the original problem and not to affect the correctness of other parts

2.4.1.4. Backlog

The sprint during which the bug was found. If it becomes impossible to fix the bug during the current sprint, it can be later moved to another sprint backlog.

2.4.1.5. Iteration goal

This should be leaved empy because it does not offer us anything useful.

2.4.1.6. Priority

The priority of a backlog item in Agilefant depends on the bug severity. The mapping is shown below.

Bug severity	Agilefant backlog item priority
blocker, critical	+++++
major	++++
normal	+++
minor, trivial	++

2.4.1.7. Responsible

The finder of the bug makes the first descriptions and should therefore be initially responsible for getting things started. The responsibility can later be delegated to someone more familiar with the problem.

2.4.1.8. Description field (CHANGE LOG)

Description field contains the history of the bug. It consists of log entries for which the format is strictly defined. An entry consists of two or more pieces of tagged information and an entry description. It looks like this:

DATE: xx.xx.xxxx yy:yy USER: [Name of the person making an entry] [TAG1]: ... [TAGn]: [Entry description]

In addition when finding a new bug the first entry should contain a description for the bug and should be as accurate as possible.

When editing anything in the backlog item here are the tags to use:

- BACKLOG CHANGE: [previous backlog] -> [new backlog]
- RESPONSIBILITY CHANGE: [previous person] -> [new person]

- STATUS CHANGE: [previous status] -> [new status]
- PRIORITY CHANGE: [previous priority] -> [new priority]
- MERGED TO: [Another bug] (see below in guidelines for explanation)

Every time the backlog item is edited, make a new log entry. Only exception to this is changing only the time estimate. The log entries should be separated with lines, sorted by time and the latest one should be on top. It is important to remember that previous entries should not be edited or deleted because they contain the history of the bug.

2.5. After each sprint

2.5.1. Collecting data

After each sprint we will collect data from all users. Data collection will be done in the form of a questionnaire.

Questions for users:

1. How often did you use Agilefant? (Select one)
   - daily
   - every couple of days
   - weekly
   - every couple of weeks
   - not at all
2. Was Agilefant useful? (1-5, 1=not at all, 5=very useful)
3. How did Agilefant benefit you? Mention at least two things.
4. How big was the extra effort required by using Agilefant? (1-5, 1=not at all, 5=too big)
5. What was bad in Agilefant? What could be changed to make it better? Mention at least two things.
6. Did Agilefant help in measuring the sprint progress? (1-5, 1=not at all, 5=very much)
7. Did Agilefant help getting an overview of the project (e.g. what others were doing)? (1-5, 1=not at all, 5=very much)
8. Other comments?

The realized benefits are asked from everyone with the questionnaire. To measure the required additional effort we'll ask about it too. We'll also try to get some usage data out of Agilefant, if at all possible. The usage data may help us to determine the effort and time it takes for people to use Agilefant. The people in Team Maranello will probably help us with data acquisition.

Other costs of using Agilefant include training, installation and maintenance. This data will be monitored by us during the project.

2.5.2. Updating Agilefant

We will update Agilefant to a new version as soon as one is available from Team Maranello. Upgrades will be done only between sprints, so that development during sprints won't get disturbed by software upgrades.

3. Experience and changes

3.1. Project planning iteration (PP)

3.1.1. Installation

Installation was done on Wednesday the 16th of October on a server provided by our customer. The installation took about 6,5 hours, including planning. That installation time includes the effort required to install the server operating system, all the components required by Agilefant and Agilefant itself.

Agilefant was installed on a server running CentOS 5.0 [CentOS] . As the Linux distribution flavour was new to us, installing Java6, which is required by Agilefant, was a new experience. But with a little studying with Google [Google] , problems were solved and Sun JDK 6.0 update 3 [SunJava] was installed on the server.

Something interesting happened during the installation of MySQL [MySQL] . We used the MySQL version that comes with CentOS. The MySQL GRANT commands described in Agilefant installation instructions worked fine without problems, but logging in the database failed.

The problem was solved by editing the mysql.user table manually. We replaced the host wildcard % with localhost for user agilefant. After that and FLUSH PRIVILEGES; the user could log in to the database. We suspect that this problem has more to do with the default configuration of CentOS or MySQL than with the installation instructions of Agilefant.

The version of Apache Tomcat [Tomcat] application server that comes with CentOS is configured for GCJ [GCJ] . Tomcat didn't work with Sun's Java6 at all. With a little googling we found out that this is a known issue. We installed Tomcat v5.5.25 by hand and Agilefant began working immediately.

3.1.2. Usage

Agilefant was not used in this iteration.

3.2. First implementation iteration (I1)

3.2.1. Usage

Agilefant was used in this iteration for the first time. Backlog items were added to the iteration backlogs and progress was updated and monitored as planned. Before the first sprint a training session was held. In the training session the concepts and usage was explained. After the training session everyone could start using Agilefant.

There were no major issues in Agilefant during the iteration. The general consensus was that Agilefant helped monitoring progress considerably. The burn-down graph was an easy way to check the sprint progress quickly. Agilefant worked as a distributed to-do list for the sprints and generally fulfilled its task well.

3.2.2. Found problems and improvement ideas

Some minor problems were found during the iteration. The problems are listed here.

- Characters ä and ö don't work in user names.
- Agilefant may crash with a `NullPointerException` if the user has been idle for a long time.
- When using Agilefant over a cellular packet data connection, the iteration page loads much more slowly than a static HTML page of similar size.

Also some ideas for improving Agilefant were found. It would be nice, if:

- Agilefant could use the full display width on a wide screen monitor.
- the state of a backlog item could be changed on the iteration page (like the states of tasks).
- backlog items could have dependencies. The dependencies must be ready before a backlog item can be implemented.
- deadlines could be defined for backlog items.
- Agilefant would inform the user after a backlog item has been saved successfully.
- Agilefant could list the amount of work allocated for each developer.
- the default sort order of backlog items could be selected, and Agilefant could remember the selection (per-user).

3.2.3. User feedback

After the first sprint user feedback was collected with a questionnaire. The user feedback is listed here.

3.2.3.1. How often did you use Agilefant?

- daily: 1 (20%)
- every couple of days: 4 (80%)
- weekly: 0 (0%)
- every couple of weeks: 0 (0%)
- not at all: 0 (0%)

3.2.3.2. Was Agilefant useful? (1-5, 1=not at all, 5=very useful)

- 1: 0 (0%)
- 2: 0 (0%)
- 3: 2 (40%)
- 4: 3 (60%)
- 5: 0 (0%)

3.2.3.3. How did Agilefant benefit you?

- It makes it easier to see how things are progressing. (or at least it would, if all the people would use agilefant as we had agreed. :D) It is light weight, it does not ask stupid questions like "are you sure you want to change that description etc.."
- Makes planning easier. Makes progress evaluation easier.
- As a listing of things to remember. As a view of progress.
- I don't think it benefited me personally, but following the project probably became easier.
- It helped to see the project status (which tasks/backlog items were done and which were not started at all). The burndown graph is nice.

3.2.3.4. How big was the extra effort required by using Agilefant? (1-5, 1=not at all, 5=too big)

- 1: 1 (20%)
- 2: 4 (80%)
- 3: 0 (0%)
- 4: 0 (0%)
- 5: 0 (0%)

3.2.3.5. What was bad in Agilefant? What could be changed to make it better?

- There should be a way to print all the backlog items and their subtasks in one screen.
- The session handling should remeber what sorting was used for backlog items.
- There should be some way to see how much time has been spent, compared to the estimate how much is left.
- Agilefant should be able to tell how many hours have been planned for each user in a sprint, based on the assigned backlog items and tasks. (this info is in the system, it would just need an easy way to report it.)
- The session timeout should be longer, now you have to log in all the time, if you keep it open in the browser...
- No dependensies for tasks. No subtasks. Better ability to define the view, I.E. sorting, different search criteria for tasks/hours etc.
- There is no indicator that the current view has actually been saved, when you push "save"-button. The best way could be returning to the previous view (one step up hierarchically). Example: Choose sprint -> choose backlog item -> choose task -> save (returns automatically to backlog item-view)
- Default sorting in views containing a list of items should be sorted alphabetically.
- There should be a way to mark a deadline for a backlog-item. When the item is created it should automatically be set to the date of the end of the iteration but it could be manually set to some other date.
- The apparent random order of the backlog items, and if you rearrange them, it doesn't stay that way when the page reloads.
- There's no possibility to give tasks subtasks etc...
- You can't "open" the backlog item in the frontpage to see it's tasks there.
- When you edit a goal or anything else there's no feedback that you have accomplished in doing it, the view stays the same.
- The backlog item moving button is only at the bottom of the list. It should also be at the top to minimize scrolling.
- When there are tasks under a backlog item and everything is marked as done, there is still this slight white part in the right end of the progress bar, this looks as though the backlog item is not completely done.
- The positioning of the SAVE and DELETE buttons is somewhat random looking. Also a Cancel-button could be there to take you back from the edit page.

- Sorting the backlog items. Could it be possible for the user to define some sorting method as default? Some deadline-tracking system would also be worth considering. (deadlines could be seen in the same picture with the burn down graph or in some other picture?)

3.2.3.6. Did Agilefant help in measuring the sprint progress? (1-5, 1=not at all, 5=very much)

- 1: 0 (0%)
- 2: 2 (40%)
- 3: 2 (40%)
- 4: 1 (10%)
- 5: 0 (0%)

3.2.3.7. Did Agilefant help getting an overview of the project (e.g. what others were doing)? (1-5, 1=not at all, 5=very much)

- 1: 0 (0%)
- 2: 3 (60%)
- 3: 2 (40%)
- 4: 0 (0%)
- 5: 0 (0%)

3.2.3.8. Other comments?

- The priority marked as "+" isn't that informative. High-Low scale would be better.

3.3. Second implementation iteration (I2)

TBD

3.4. Conclusions

TBD

References

[[ScrumAlliance]] *http://www.scrumalliance.org/* .

[[Agilefant]] *http://www.agilefant.org/* .

[[MaranelloSEPA]] *http://www.agilefant.org/wiki/display/PRO/SEPA* .

[[Blog]] *http://agilefant.blogspot.com/* .

[[CentOS]] *http://www.centos.org/* .

[[Google]] *http://www.google.com/* .

[[SunJava]] *http://java.sun.com/* .

[[MySQL]] *http://www.mysql.com/* .

[[Tomcat]] *http://tomcat.apache.org/* .

[[GCJ]] *http://gcc.gnu.org/java/* .