**The Report committee for Anil Kumar Bondalapati**
**Certifies that this is the approved version of the following report:**

**A Comparative Evaluation of Project Management Tools for Extreme Programming**

**APPROVED BY**
**SUPERVISING COMMITTEE:**

**Supervisor:** ______________________________
Dewayne, Perry E

______________________________
Krasner, Herb

# A Comparative Evaluation of Project Management Tools for Extreme Programming

By

**Anil Kumar Bondalapati, BEE**

**Report**

Presented to the Faculty of the Graduate School

Of the University of Texas at Austin

In Partial Fulfillment

Of the Requirements

For the Degree of

**Master of Science in Engineering**

The University of Texas at Austin

May, 2012

**A Comparative Evaluation of Project Management Tools for Extreme Programming**

**By**

Anil Kumar Bondalapati, MSE
The University of Texas at Austin, 2012
SUPERVISOR: Dewayne, Perry E

Extreme Programming, which is commonly termed as XP, is considered to be the most widely used “agile” software methodology among all that are available. As it is a kind of agile software development, XP promotes regular releases in brief development cycles and that is meant for improving efficiency plus introduction of checkpoints where the latest client necessities can be implemented. This research study evaluated the project management tools that are commonly used in the industry to support an XP development team. A selected set of these tools have been evaluated and their strengths and weaknesses are reported here. We also performed a gap analysis on all the tools that are available in the market. We determined the three best tools that are widely used and also to what extent these tools support the XP practices. One major conclusion from the research is that there is no single tool available that will help with all the various types of XP projects. The three tools that are most widely used cover most of the XP projects but still they have some limitations in the areas of communications management, human resources management and risk management. As a result of these results our next steps will be doing more research on PPTS tool and introduce the missing features as this is an open source tool. We also would like to do more research on the development, testing tools that can be plugged into this tool to come up with a single tool to work with the XP projects.

## Table of Contents

## List of Figures

## List of Tables

## 1. Introduction to Extreme Programming

Extreme programming is a software development discipline that focuses on improving software quality by stressing the changing customer requirements. It is intended to release software in short development cycles which results to improve productivity and allows customer to introduce new checkpoints where new requirements can be adopted **(Auer and Miller, 1996)**.

Extreme Programming emphasizes on teamwork. It works by bringing whole team together by enduring simple practices and feedback from everyone that enables the team to estimate their present situation and possible future prospects.

Different companies all over the world have seen many benefits from XP. The main reason for its success is because of its concentration on customer satisfaction and necessities. Extreme programming allows the developers to fulfill the latest customer needs even in the late life cycle. It basically emphasizes on the team work and customers. Managers and developers are all equivalent partners in the joint effort.

## 2. Values Relied By Extreme Programming

Extreme Programming improves a software project by relying on five values:

1. **Communication:** Communication is the chief factor towards quick development and client approval. XP exemplifies communication with its goal on simplicity, which is use of person-to-person communication rather than written documents whenever and wherever probable. It also centers on repeated communication between the client and development group members by keeping an on-site customer as development advances. This on-site customer judges what has to be built and in the order in which it needs to be built. **(Cohn's, 2000)**.

2. **Simplicity:** Simplicity implies that the software is designed by employing the simplest possible construction and design, although it has a deeper meaning. Simplicity permeates the whole process. Instead of trying to make all code modules very flexible in anticipation of future requirements, Extreme Programming focuses on building reliable and simple code to satisfy the requirements of the work packages that have already been authorized. As new features are added, or stories are authorized, the design is reworked and modified to accommodate the most recently scheduled features. One of XP‟ s rules is to add a process or software only when it is really needed and there is no software built with the expectancy of need **(Beck and Fowler, 1998)**.

3. **Feedback:** The team gets constant feedback from the testing since inception. The customer is also involved since the beginning of the project. The team delivers the system to the customer in short releases and gets feedback from the customer **(Beck and Fowler, 1998)**. The team then implements changes in the system as suggested in the feedback from the customer.

4. **Respect:** Success of each release deepens the respect for contributions of each and every team member **(Cohn's, 2000)**.

5. **Courage:** The team courageously responds to the changing customer requirements **(Auer and Miller, 1996)**.

## 3. XP Practices

- **The Plan**

  The business as well as the development teams get together in order to decide the features of the required system which will be of maximum value to the business. Extreme Programming teams capture Work Packages as hand-written descriptions of user functionality often written on an index card. These index cards are called Story Cards to remind project participants that the contents of each card should describe a story that a user might tell about functionality instead of the technical activities that software developers instinctively describe. The cards are never formalized; and there is no identification of any relationships or dependencies between the cards that are being spotted out. Software developers place time estimates and the customers assign priorities to each card [Jacobson 1992]. Together, the developers and the customers play the “Planning Game" in which the customer chooses those User Stories that comprise the most important substance for a small, incremental deliverable of approximately a month. Every small execution increment is being acknowledged and tested by the client. After that, the left over User Stories are re-checked for probable necessity and/or priority changes and the Planning Game is re-played for the next implementation increment.

- **Short Releases**

  A simple system comprising of a valuable set of characteristics is made early on and updated regularly in small cycles. XP increases the rate of coiled progress by having diminutive releases of 3-4 weeks. Following the end of each release, the client reviews the temporary product, identifies the flaw, and adjusts the future requirements.

- **Symbol**

  Every project has an array of names as well as a description that aids in guiding the growth process and the interaction between all the people. In XP it is ensured that each application should have theoretical integrity on the basis of a simple symbol that explains the real meaning of how the system works.

- **Simple Form of Design**

  The making of an application is always done by taking a simple design because it helps to meet the present business requirements. The future of the application can be thought of later because the requirements keep on changing. Refactoring practice is done in order to ensure that the design meets a high standard. XP aims for extremely simple designs.

  The methodology emphasize on the fact that programmers should not be busy predicting the future needs and eventually generate more complex designs .

Rather developers must follow a simple design practice so that it makes XP give the maximum possible output.

- **Tests and Trials**

  XP makes sure that sufficient tests are done prior to adding new traits. Before adding new features, tests are performed in order to validate the software. The software is designed and made with the capacity to pass these tests. Any software that is developed with XP is verified at all times. Generally, two forms of testing are conducted, namely unit testing and functional testing.

- **Unit Testing**

  Wide-ranging, automated tests are conducted before the production code is generated. In order to satisfy the requirement for 100% of all quality control tests to pass after each incremental coding step, the need for those tests to run in an automated test harness becomes a necessity. Each and every public method or function must have a functional specification captured in the form of an automated test suite. These automated unit tests are considered to be such an important part of the design process, they are required to be written before the code they are intended to test is written. This makes sure that the new code executes the new feature without breaching anyone else’s code.

  Unit Tests specify two key actions:

1. In order to ensure that new aspects function, Unit Tests are written for every aspect. They must be written before the code is written and all the Unit Tests are saved for the entire system.

2. In order to ensure that the other functionality not is broken, all the Unit Tests in the whole set up is run before the release of any code, and it is also ensured that those tests are running at 100%. If in case something breaks, the exact location of the problem can be known. When every Unit Test in the whole setup is running 100%, it not only indicates that the new feature is functioning, but also that nothing has broken anywhere.

- **Functional Test**

  Conventionally, project managing techniques have been based on the developers own assessment as to how much of the job has been fulfilled. As an alternative, XP emphasizes the use of tracking by functional test case for calculating the project entirety. This type of evaluation is known as "Project Velocity" in XP. Functional test cases are made on the basis of client scenarios. When a functional test case is effectively passed, it can be said that a particular feature has been executed accurately. The completeness of the project is calculated on the basis of the percentage of functional test cases that have passed. This measure can be calculated unambiguously by the team members.

Here are the main features of Extreme Functional Tests:

1. **Client-owned:** clients must comprehend the tests for deriving confidence from them.

2. **All-inclusive:** clients must select values for which the test would hold meaning in case it succeeds or fails.

3. **Repeatable:** developers must write new tests when a new increment is being run.

4. **Automatic:** There must be a testing facility made by developers for setting up and running the tests, checking their results, and reporting them.

5. **Timely:** The tests need to show up regularly after the end of development or else it just drags things down.

6. **Public:** clients require checking the software development progress; hence a turn down in test scores should not be neglected by the developers.

- **Refactoring**

  Refactoring is the process by which the codes structure can be improved, while keeping its function intact [Jeffries et al. 2001]. XP makes use of the refactoring code constantly and unambiguously. It is a technique that improves the design of an already existing codebase. Its fundamental nature is to apply a sequence of small behavior preserving the transformations that improve the structure of the code. By doing these in short steps, it helps to reduce the risk of errors [Fowler and Martin 2003].

- **Pair Programming**

  Programmers making use of XP are always paired up and they write all the production codes by using one machine per pair. This activity helps the code to be continuously reviewed while being written. Pair Programming has proven to generate high quality code with little or almost no reduction in productivity [ManLui et al. 2003].

- **Joint Code Possession**

  All the codes belong to each and every member of the team; therefore no single team member can own a piece of code. Furthermore anyone can make alterations to the code base at any time. This motivates everyone to contribute innovative ideas to all the sections of the project.

- **Constant Integration**

  **S**oftware systems are made and integrated many times a day. At the very least all alterations are integrated into the main code base on an integration machine, at least one time per day. Thus, there are many products that are built per day. Each product that is built is tested using the coupled test cases.

- **40-Hour Week**

  Programmers in an XP project normally stick to a 40 hour week of working in order to retain output and evade burn outs. It was found that during critical times when overtime is on the rise, the artifacts that are generated are poor.

- **On-Site Consumer**

  One or more patrons who will use the system that is being built are passed on to the development team. The client helps in guiding the development process and has the power to prioritize the necessities and answer any queries the developers may have. This ensures that there is effective interaction with the customer and as a result fewer documents will be required.

- **Coding Standards**

  Anyone involved in an XP project uses the same Coding Standards which makes it easy for working in pairs and sharing must be definite and followed. This is very important during the project to maintain the standards to maintain the quality of the product.

**Extreme Programming**

Extreme Programming is a new approach to software development which takes a code-centric view of the activity.

XP is different from traditional approaches in many ways:

1. XP involves short, tight iterations of building and releasing software. In our case study company, these iterations were 3 weeks long, but others have used 2 week long iterations. At the end of this time something new has been released.
2. Requirements are gathered in terms of 'stories'. These usually begin with 'We want…' (Requirements or stories???) are produced by the customer or client. The developers look at the stories and estimate how long they think it will take to satisfy them. Developers and customers negotiate what requirements can be delivered in the time available. This exposes the customer to developer’s estimates, and encourages them to prioritize their requirements since it is not possible to do everything in a single iteration. The activity of estimating and prioritizing to see what can be done in the next iteration is called the planning game.
3. The customer is on site and is part of the development team. Once the iteration has started, developers work in pairs. No development is done by a single individual. All development is jointly owned. This is a surprisingly successful element of XP.

4. A 'test-first' philosophy prevails. In XP one writes the test to show that the code will work before the code is written. Testing and coding are performed together. Before any new software is released, the whole test suite is run to ensure that it does not break any other component of the system.
5. Some principles underlying the people aspects of XP are: work no more than 40 hours a week, be courageous, take responsibility and share ownership.
6. Some principles underlying code production are: keep it simple, have one shared metaphor to guide the system development, regularly restructure the system to improve it (refactoring), continuously integrate and test, and follow coding standards.

**4. Extreme Programming Model**

The most important model of pair programming is:-

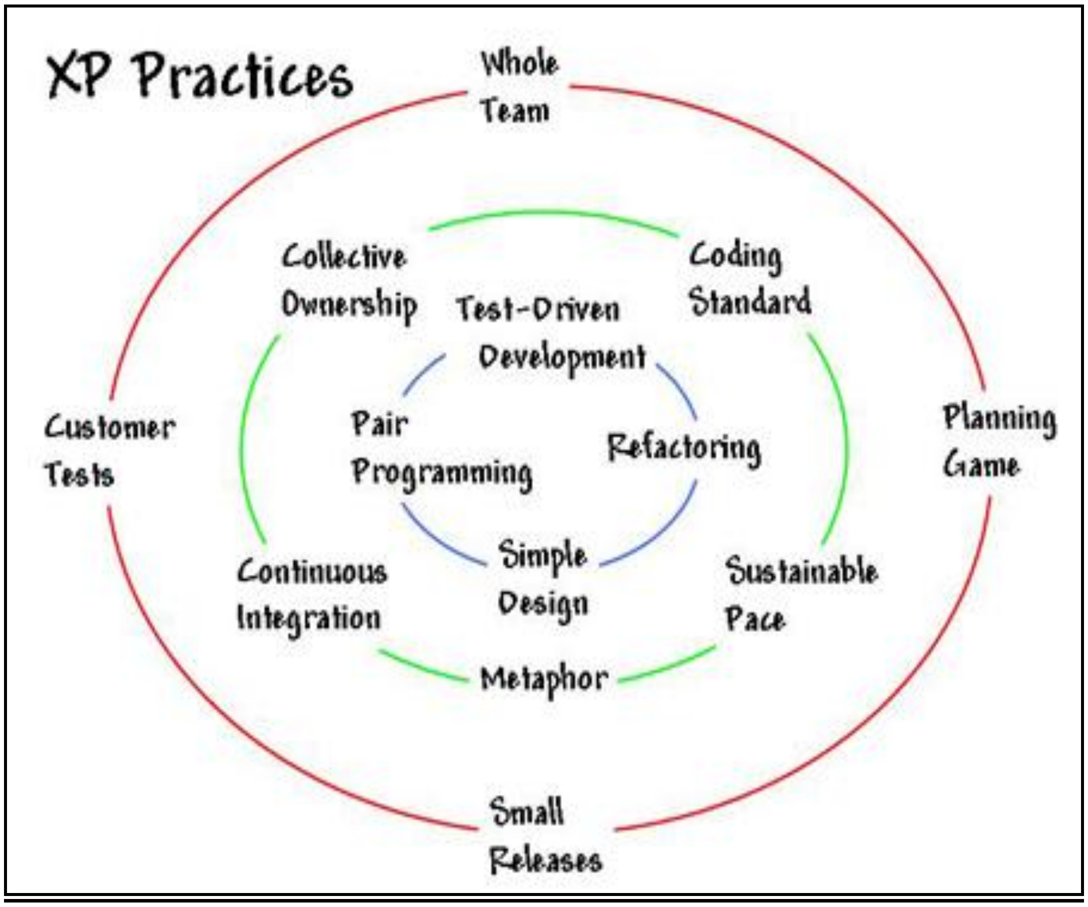


**Figure 1 Extreme Programming Diagram**

**<u>Agile Development Model:</u>**

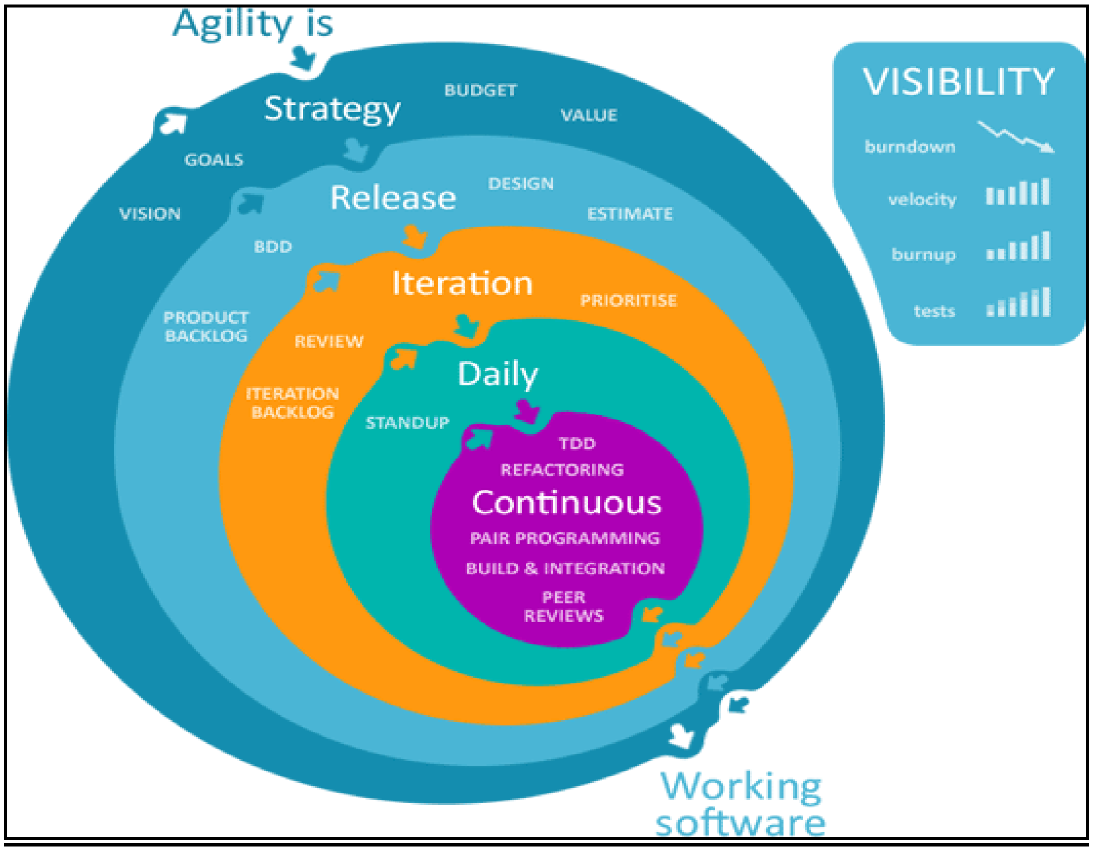


**Figure 2 Agility Development Model**

## Extreme Programming Methodology Flow

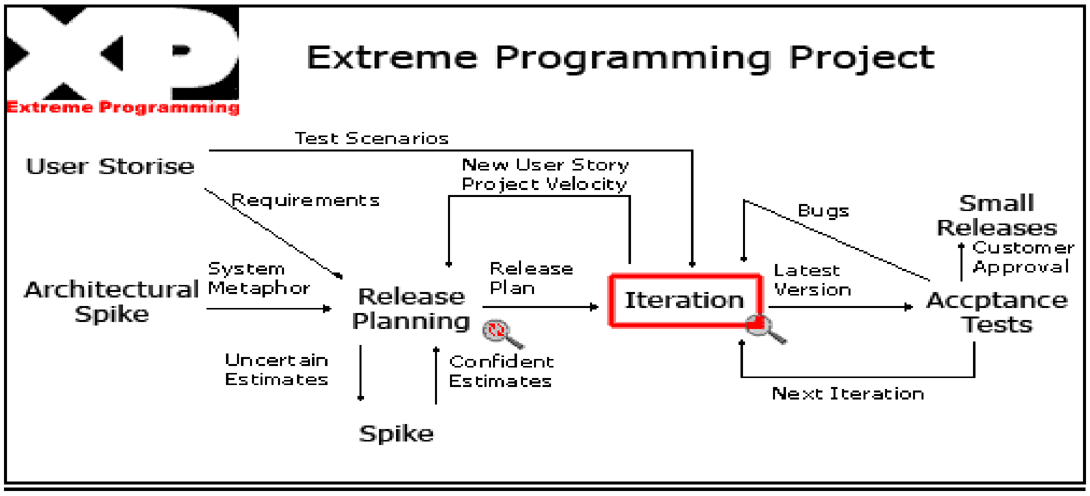


**Figure 3 Extreme Programming Methodology Flow**

## 5. Famous Projects Which Implemented Extreme Programming

The most famous projects which implemented extreme programming are:

1. **Chrysler Comprehensive Compensation System (C3) Project:** C3 was a project in Chrysler Corporation to replace several payroll applications with a single application. Extreme programming was used in this project for the first time.

2. **Primavera:** The twenty one year old organization changed its entire working around extreme programming.

3. **Symantec Corporation:** Symantec Corporation declared that many of its teams are transited to extreme programming.

4. **Fidelity Investments:** Fidelity Investments largest project that deals with all the financial transactions of the companies was built around extreme programming in eight months.

## 6. Activities of the Extreme Programming

The four basic activities that are performed within the Extreme Programming software development methodology are:

1. **Coding:** Coding is done by the programmer and other programmers can give feedback on that code **(Cohn's, 2000).**

2. **Testing:** Extreme Programming testing teams perform Unit test and Acceptance test. A tester tests every piece of code in Unit testing. Acceptance testing verifies that the requirements as understood by the programmers and match the customer's actual requirements **(Cohn's, 2000)**.

3. **Listening:** Programmers need to listen to the customer's requirements and their business logic **(Cohn's, 2000)**.

4. **Designing:** The team should implement a designing structure that organizes the different logics of the system **(Cohn's, 2000)**.

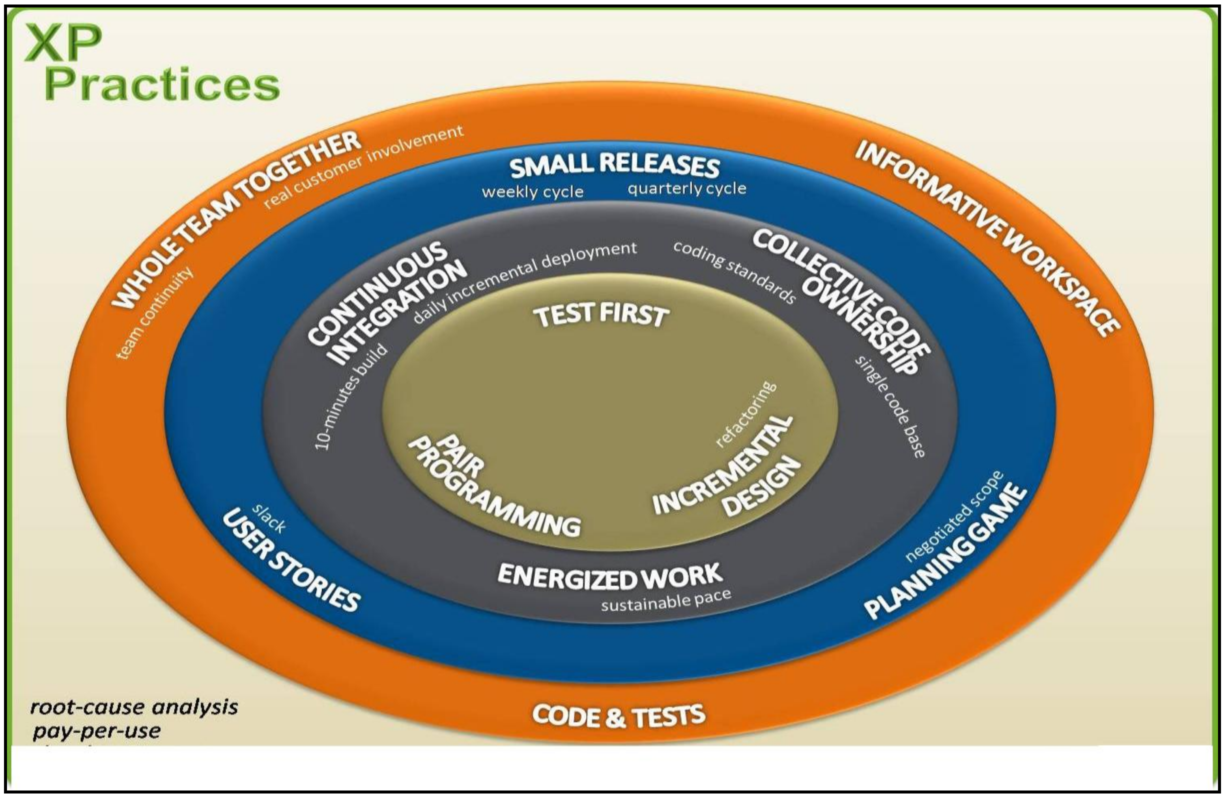


**Figure 4 Extreme Programming Practices**

## 7. Tools Available In Market

The tools available in the market for Extreme Programming are as follows:

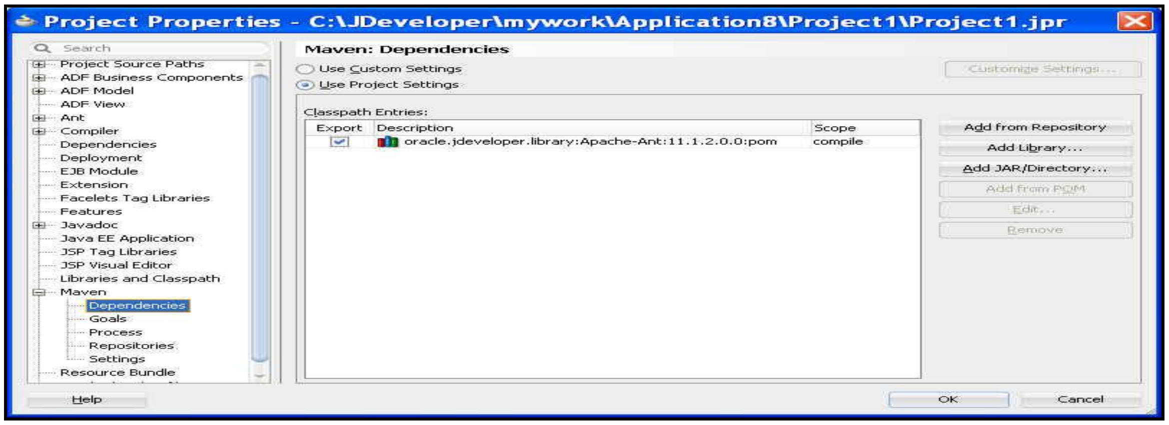
1. Apache Ant (Java Extreme Programming Tool).
2. Project Engine.
3. Rally Agile.
4. TargetProcess.
5. TrackIT.
6. Extreme Planner.
7. XPWeb.
8. XPMT (Extreme Project Management Tool).
9. XPML (Extreme Programming Markup Language).
10. XDoclet.
11. VersionOne.
12. XPlanner.
13. Project Planning And Tracking System (PPTS)
14. Extreme Management Tool.

## 1. Apache Ant:

Apache Ant is an extreme programming tool for automating software development processes. It is implemented in Java language and requires a Java platform. It is best fitted to develop Java projects from scratch.  Apache Ant uses XML to identify its dependencies and build processes. The most important benefit of Apache Ant is that it resolves portability issues. It has a large amount of built-in functionalities as it is based on XML and so it is implemented in cross platform **(Beck and Fowler, 1998)**.

### Limitations of Apache Ant:

- It has a very complex structure and the scripts of the tool are very large.  This tends it to be very slow.
- The global property once created in Apache Ant cannot be changed at all.
- It has faulty user handling rules and it is stateless so cache files are not saved at all.


Project Properties - C:\JDeveloper\mywork\Application8\Project1\Project1.jpr

Maven: Dependencies

( ) Use Custom Settings
(•) Use Project Settings

Classpath Entries:

| Export | Description | Scope |
|---|---|---|
| ✓ | oracle.jdeveloper.library:Apache-Ant:11.1.2.0.0:pom | compile |

**Figure 5 Apache Ant**

## Advantages of Ant

The five advantages of using Ant are:

- Extensibility: Use Ant with various cross-platform Java classes. It works with JavaScript, NetRexx, and Python. Even use the simple Ant exec task to call shell scripts and operating system commands, and executables. Write a cross-platform version of the task in Java or with the Ant scripting task.
- Documentation: Ant includes a script that automates and documents the build-and-deploy process.
- Continuous integration: You can use Ant to integrate the units, such as Java classes of an application. Ant runs the units based on their dependency on other units. This reduces development time because we need to specify the dependency in a build file. Every time Ant is run, it automatically determines the dependency.
- Automated testing: Ant simplifies the building and testing software by enabling developers to perform automated testing as often as required.
- Standard for automating Java builds: Ant is used by various applications, application servers, and Integrated Development Environment (IDE), such the Sun Java 2 Enterprise Edition (J2EE) blueprint applications, the Orion application server, the projects at Apache Jakarta, the Net Beans IDE, and Together Control.

## Continuous Integration with Ant

The four basic components of Ant are:

- Tasks: These are functions such as copying files, compiling files, and creating JAR files.
- Targets: These are a collection of tasks that you want to execute. A target can depend on other targets. Use the depends attribute to indicate that one target depends on another. A target "A" depends on another target "B". Target A does not execute until target B is executed.

**Standard Ant Targets**

You can use any alphanumeric string as a target name. You can also use hyphens with the target names, but targets with hyphens cannot be specified from the command prompt. You should use hyphens only for internal targets. Internal targets are not specified at the command prompt and are used as intermediate targets.

**Tip** Although you can use special symbols, such as space and comma, in target names, avoid these symbols because they will not be supported in future versions of Ant.

You can use any name to specify a target, a few names are used to represent frequently performed set of tasks. Table 2-1 describes the standard Ant target names.

**Table 1 Target Names**

| Target Name | Description |
|---|---|
| Test | Runs the unit tests. |
| Clean | Cleans the output directories. |
| Deploy | Ports the Java Archives (JAR) and Web Archives (WAR) to the execution system. |
| Publish | Provides the output of the source and binaries to any distribution site. |
| Fetch | Gets the latest source from the Concurrent Versioning System (CVS) tree. |
| docs or Javadocs | Provides the outputs documentation. |
| All | Performs clean, fetch, build, test, docs, and deploy. |
| Main | The default build process. |

**Table 2 Internal Targets**

| Target Name | Description |
|---|---|
| Init | Initializes properties in per-user property files and performs other initialization tasks. |
| init-debug | Initializes debug properties. |
| init-release | Initializes release properties. |
| Compile | Performs the actual compilation. |
| link/jar | Makes JAR or equivalent files. |
| Staging | Executes any pre deployment process before moving on to the production site. |

**Conditional Targets**

The conditional execution of targets is achieved by two optional attributes, “if” and “unless”. The condition is based on whether or not a property is set, regardless of the value of the property. For example, in this command, the target deploy_release is executed only if the property release is set:

```
<target name = "deploy_release" if = "release" >
```

In the following command, the target deploy_internal is executed only if the property release is not set:

```
<target name = "deploy_internal" unless = "release" >
```

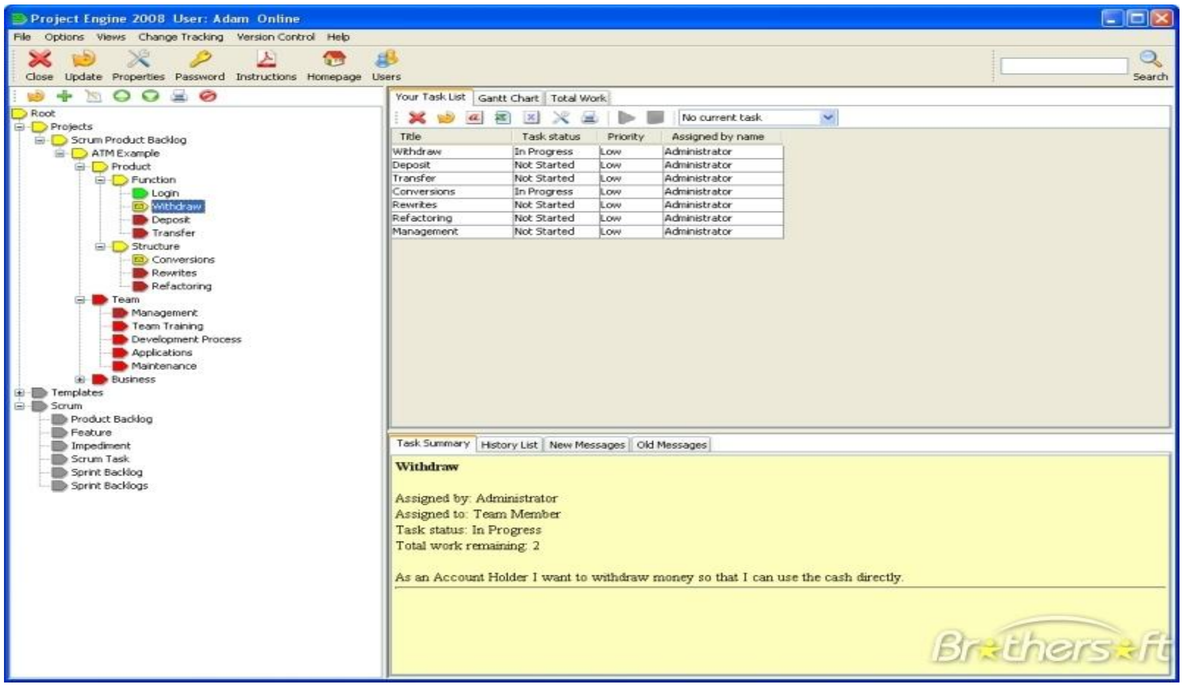
## **2. Project Engine:**

Project Engine is a project management tool available for free in today’s market. Project Engine is available both as a standalone application as well as a distributed server application. It includes all the features of project management tools like Gantt charts, time reporting, scheduling, and workload reports It is easily available on the internet so different team members can connect to it and work on all the aspects of extreme programming tool.

It creates and manages tasks and also creates public and private privileges of these tasks to all the team members. Project Engine tool is used for project planning since it tracks the work down to the smallest activities. It has a simple design, quick installation and a simple configuration as it increases productivity **(Wake, 1997)**.

**<u>Limitation of Project Engine:</u>**

- ➢ Project Engine has a small learning curve so it lacks enhancement.



| Title | Task status | Priority | Assigned by name |
|---|---|---|---|
| Withdraw | In Progress | Low | Administrator |
| Deposit | Not Started | Low | Administrator |
| Transfer | Not Started | Low | Administrator |
| Conversions | In Progress | Low | Administrator |
| Rewrites | Not Started | Low | Administrator |
| Refactoring | Not Started | Low | Administrator |
| Management | Not Started | Low | Administrator |

**Withdraw**

Assigned by: Administrator
Assigned to: Team Member
Task status: In Progress
Total work remaining: 2

As an Account Holder I want to withdraw money so that I can use the cash directly.

**Figure 6 Project Engine Tool**

**<u>3. Rally Agile:</u>**

Rally Agile tool is used to plan agile projects from scratch to the releases and all the iterations. It is based on the ASP model and the data and application is hosted from a remote location. It supports software development by using Agile ALM products. It also has an integration and customization feature. The distributed teams work well together as it works on Agile ALM processes. Rally Agile is an enterprise proven agility tool. The

Rally platform includes modules for project management, time tracking, portfolio management, and requirements management. It focuses on the individual and interaction over the processes and tools, customer collaboration and responding to change over the plan **(Cohn's, 2000)**.

**<u>Limitations of Rally Agile:</u>**

- Rally Agile lacks in generating customized reports. The reports once created cannot be modified.
- Common Backlogs are not created so multiple teams cannot work together.

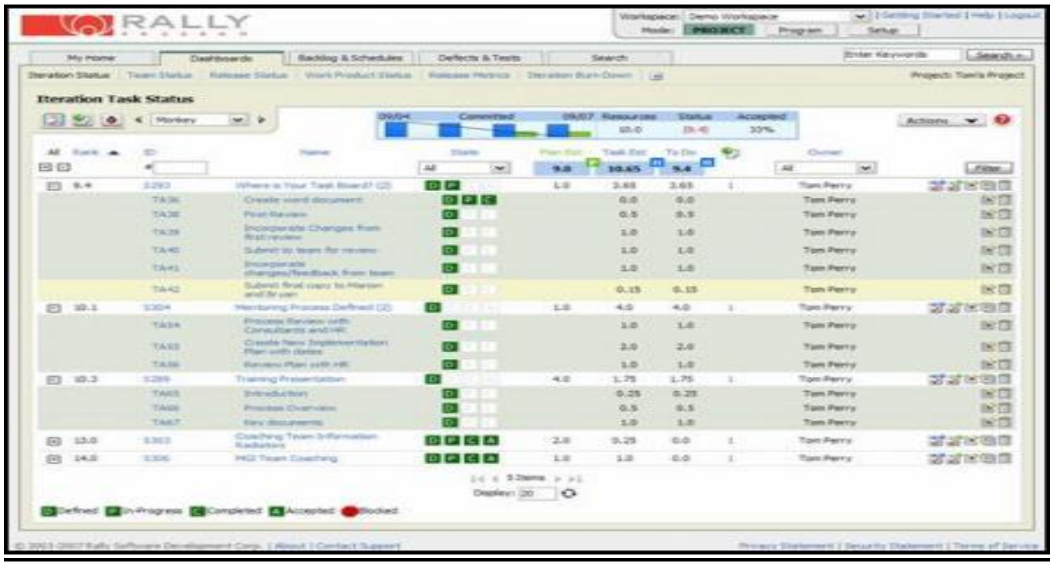


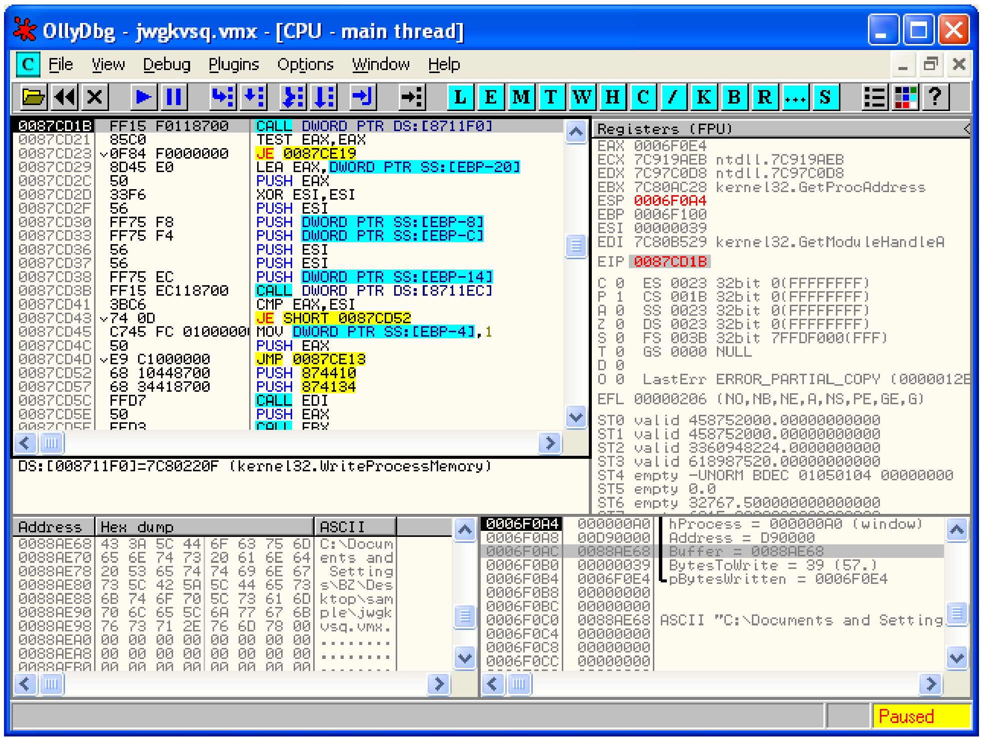
**Figure 7 Rally Agile Tool**

## **<u>4. Target Process:</u>**

Target Process is an Agile Process Management tool that automates all the tasks associated with the agile project **(Wake, 1997).** It assists in planning, tracking and quality assurance of the project. It assists in data cache, real time reports and lets the management know the status of the project. It has an iteration planning and program level release planning. It tracks a large number of projects within a single program. It integrates the testing process with the tracking process and project planning. The tool allows development process and workflows to be customized, plan iterations, releases, and creating role-based permissions. It has a feature for  tagging and searching. E-Mail notification, file sharing and commenting is also available in this tool.

### **<u>Limitation of TargetProcess:</u>**

- TargetProcess has a minimal requirement of 2 GHz. It needs an application server of IIS server 6.0.



OllyDbg - jwgkvsq.vmx - [CPU - main thread]

```
0087CD1B   FF15 F0118700    CALL DWORD PTR DS:[8711F0]
0087CD21   85C0             TEST EAX,EAX
0087CD23  v0F84 F0000000    JE 0087CE19
0087CD29   8D45 E0          LEA EAX,DWORD PTR SS:[EBP-20]
0087CD2C   50               PUSH EAX
0087CD2D   33F6             XOR ESI,ESI
0087CD2F   56               PUSH ESI
0087CD30   FF75 F8          PUSH DWORD PTR SS:[EBP-8]
0087CD33   FF75 F4          PUSH DWORD PTR SS:[EBP-C]
0087CD36   56               PUSH ESI
0087CD37   56               PUSH ESI
0087CD38   FF75 EC          PUSH DWORD PTR SS:[EBP-14]
0087CD3B   FF15 EC118700    CALL DWORD PTR DS:[8711EC]
0087CD41   3BC6             CMP EAX,ESI
0087CD43  v74 0D            JE SHORT 0087CD52
0087CD45   C745 FC 0100000  MOV DWORD PTR SS:[EBP-4],1
0087CD4C   50               PUSH EAX
0087CD4D  vE9 C1000000      JMP 0087CE13
0087CD52   68 10448700      PUSH 874410
0087CD57   68 34418700      PUSH 874134
0087CD5C   FFD7             CALL EDI
0087CD5E   50               PUSH EAX

DS:[008711F0]=7C80220F (kernel32.WriteProcessMemory)
```

```
Registers (FPU)
EAX 0006F0E4
ECX 7C919AEB ntdll.7C919AEB
EDX 7C97C0D8 ntdll.7C97C0D8
EBX 7C80AC28 kernel32.GetProcAddress
ESP 0006F0A4
EBP 0006F100
ESI 00000039
EDI 7C80B529 kernel32.GetModuleHandleA

EIP 0087CD1B

C 0  ES 0023 32bit 0(FFFFFFFF)
P 1  CS 001B 32bit 0(FFFFFFFF)
A 0  SS 0023 32bit 0(FFFFFFFF)
Z 0  DS 0023 32bit 0(FFFFFFFF)
S 0  FS 003B 32bit 7FFDF000(FFF)
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_PARTIAL_COPY (0000012B

EFL 00000206 (NO,NB,NE,A,NS,PE,GE,G)

ST0 valid 458752000.00000000000
ST1 valid 458752000.00000000000
ST2 valid 3360948224.0000000000
ST3 valid 618987520.00000000000
ST4 empty -UNORM BDEC 01050104 00000000
ST5 empty 0.0
ST6 empty 32767.500000000000000
```

```
Address   Hex dump                 ASCII
0088AE68  43 3A 5C 44 6F 63 75 6D  C:\Docum
0088AE70  65 6E 74 73 20 61 6E 64  ents and
0088AE78  20 53 65 74 74 69 6E 67   Setting
0088AE80  73 5C 42 5A 5C 44 65 73  s\BZ\Des
0088AE88  6B 74 6F 70 5C 73 61 6D  ktop\sam
0088AE90  70 6C 65 5C 6A 77 67 6B  ple\jwgk
0088AE98  76 73 71 2E 76 6D 78 00  vsq.vmx.
0088AEA0  00 00 00 00 00 00 00 00  ........
0088AEA8  00 00 00 00 00 00 00 00  ........
```

```
0006F0A4   000000A0  ┌hProcess = 000000A0 (window)
0006F0A8   00D90000  │Address = D90000
0006F0AC   0088AE68  │Buffer = 0088AE68
0006F0B0   00000039  │BytesToWrite = 39 (57.)
0006F0B4   0006F0E4  └pBytesWritten = 0006F0E4
0006F0B8   00000000
0006F0BC   00000000
0006F0C0   0088AE68  ASCII "C:\Documents and Setting
0006F0C4   00000000
0006F0C8   00000000
0006F0CC   00000000
```

C File View Debug Plugins Options Window Help
L E M T W H C / K B R ... S
Paused

**Figure 8 TargetProcess**

## 5. TrackIT:

TrackIT is a web-based project tracking tool **(Auer and Miller, 1996)**. It is designed to provide flexibility and customization. It has a built-in support for programming constructs.

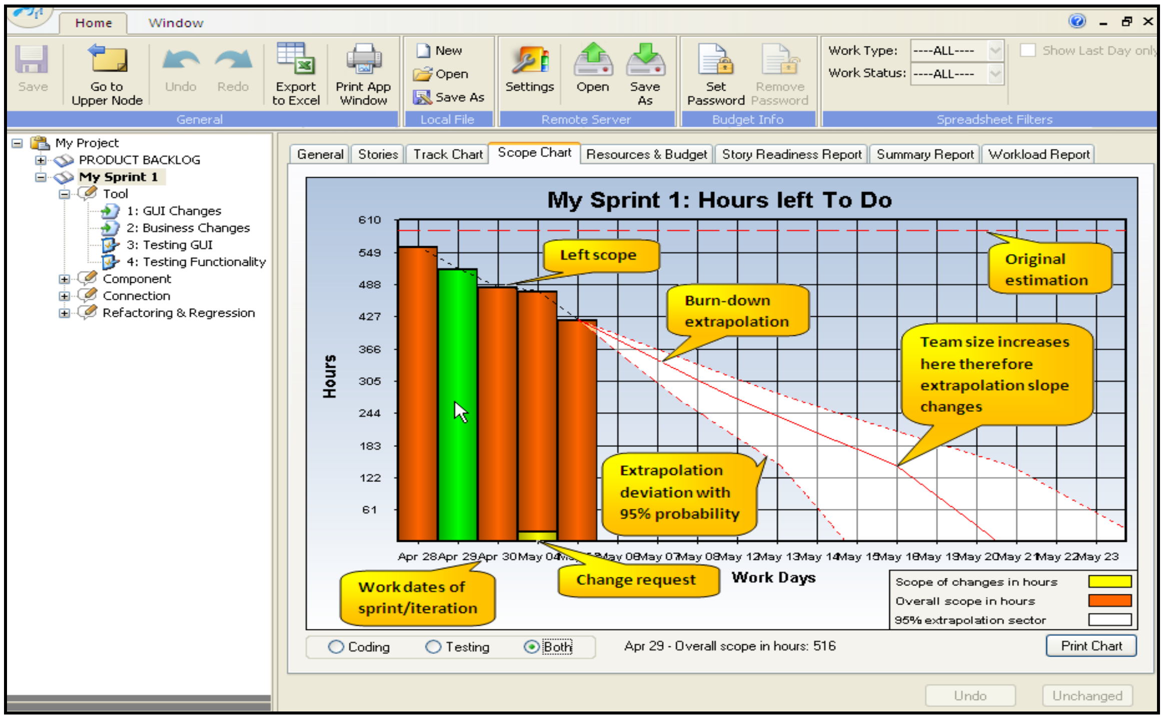


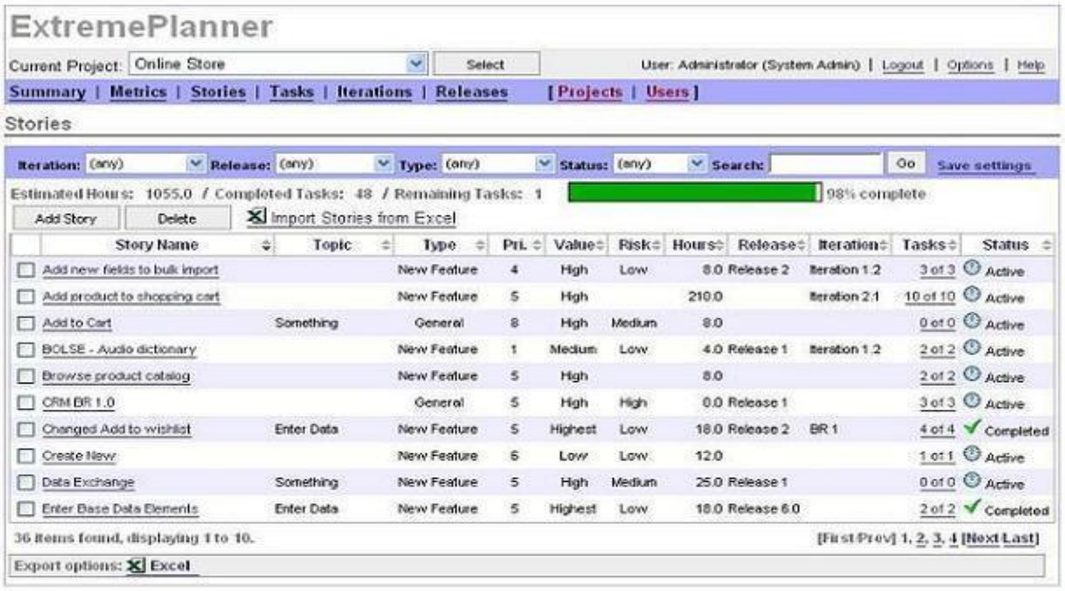
**Figure 9 TrackIT Tool**

## 6. Extreme Planner:

It is an agile project management tool for the team that operates from different locations. It focuses on task lists. It tracks the progress of features that have an actual investment value to the customers. It is a web based tool **(Cohn's, 2000)**. It is designed to support agile methods like **Scrum**. It calculates and plans releases with the drag-drop features. It manages test cases, features, defects and development in one window. It has an integrated issue tracking features.

**<u>Limitation of Extreme Planner tool:</u>**

- Extreme Planner tool lacks in customized report planning. It has no modification feature for customized reports.



ExtremePlanner

Current Project: Online Store | Select | User: Administrator (System Admin) | Logout | Options | Help

Summary | Metrics | Stories | Tasks | Iterations | Releases [ Projects | Users ]

Stories

Iteration: (any) Release: (any) Type: (any) Status: (any) Search: Go Save settings

Estimated Hours: 1055.0 / Completed Tasks: 48 / Remaining Tasks: 1 98% complete

Add Story | Delete | Import Stories from Excel

| | Story Name | Topic | Type | Pri. | Value | Risk | Hours | Release | Iteration | Tasks | Status |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ☐ | Add new fields to bulk import | | New Feature | 4 | High | Low | 8.0 | Release 2 | Iteration 1.2 | 3 of 3 | Active |
| ☐ | Add product to shopping cart | | New Feature | 5 | High | | 210.0 | | Iteration 2.1 | 10 of 10 | Active |
| ☐ | Add to Cart | Something | General | 8 | High | Medium | 8.0 | | | 0 of 0 | Active |
| ☐ | BOLSE - Audio dictionary | | New Feature | 1 | Medium | Low | 4.0 | Release 1 | Iteration 1.2 | 2 of 2 | Active |
| ☐ | Browse product catalog | | New Feature | 5 | High | | 8.0 | | | 2 of 2 | Active |
| ☐ | CRM BR 1.0 | | General | 5 | High | High | 0.0 | Release 1 | | 3 of 3 | Active |
| ☐ | Changed Add to wishlist | Enter Data | New Feature | 5 | Highest | Low | 18.0 | Release 2 | BR 1 | 4 of 4 | ✓ Completed |
| ☐ | Create New | | New Feature | 6 | Low | Low | 12.0 | | | 1 of 1 | Active |
| ☐ | Data Exchange | Something | New Feature | 5 | High | Medium | 25.0 | Release 1 | | 0 of 0 | Active |
| ☐ | Enter Base Data Elements | Enter Data | New Feature | 5 | Highest | Low | 18.0 | Release 6.0 | | 2 of 2 | ✓ Completed |

36 items found, displaying 1 to 10. [First/Prev] 1, 2, 3, 4 [Next/Last]

Export options: Excel

**Figure 10 Extreme Planner Tool**

## <u>7. XPWeb:</u>

XPWeb is a web based tool to manage extreme programming projects **(Auer and Miller, 1996)**. It includes planning and creating statistic reports. It also has a functionality that has XML export capability, authentication system, basic documentation handling and multi-project handling. It is written on PHP. It also has a calendar. It is implemented on Windows/Unix platform.

**Limitation of XPWeb:**

- It has a non-rational user interface.
- The development is suspended using XPWeb tool since 2006.
- It does not support releases and iterations.

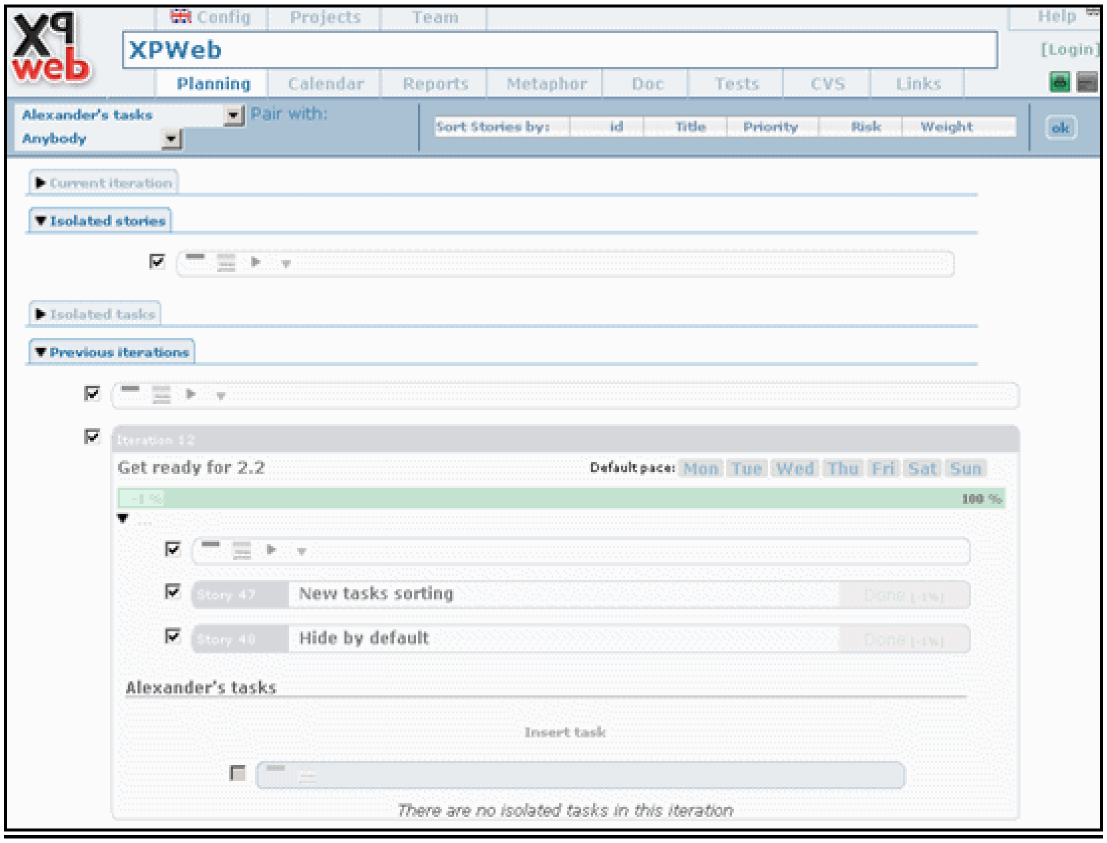
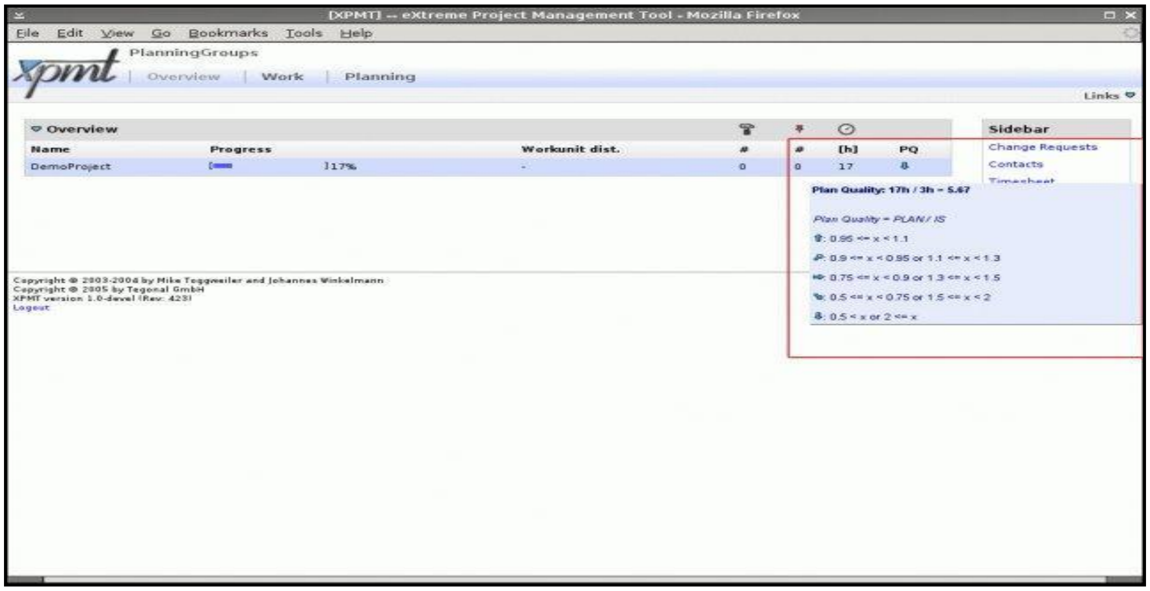


**Figure 11 XPWeb Tool**

## 8. Extreme Project Management Tool (XPMT):

Extreme Project Management Tool is a web based tool that involves project planning, controlling and bug tracking **(Auer and Miller, 1996)**. It is an open source project meant to provide a web based project tool. It includes features like work planning ability, iteration tracking, release tracking, and time planning abilities. **Limitations of Extreme Project Management Tool:**

- It has a very complex structure and processing power is very low.
- It does not customize reports and the reports lag functionalities.



▽ Overview

| Name | Progress | | Workunit dist. | # | # | [h] | PQ |
|---|---|---|---|---|---|---|---|
| DemoProject | | 17% | - | 0 | 0 | 17 | |

Plan Quality: 17h / 3h = 5.67

*Plan Quality = PLAN / IS*

: 0.95 <= x < 1.1

: 0.9 <= x < 0.95 or 1.1 <= x < 1.3

: 0.75 <= x < 0.9 or 1.3 <= x < 1.5

: 0.5 <= x < 0.75 or 1.5 <= x < 2

: 0.5 < x or 2 <= x

Copyright © 2003-2004 by Mike Toggweiler and Johannes Winkelmann
Copyright © 2005 by Tegonal GmbH
XPMT version 1.0-devel (Rev: 423)
Logout

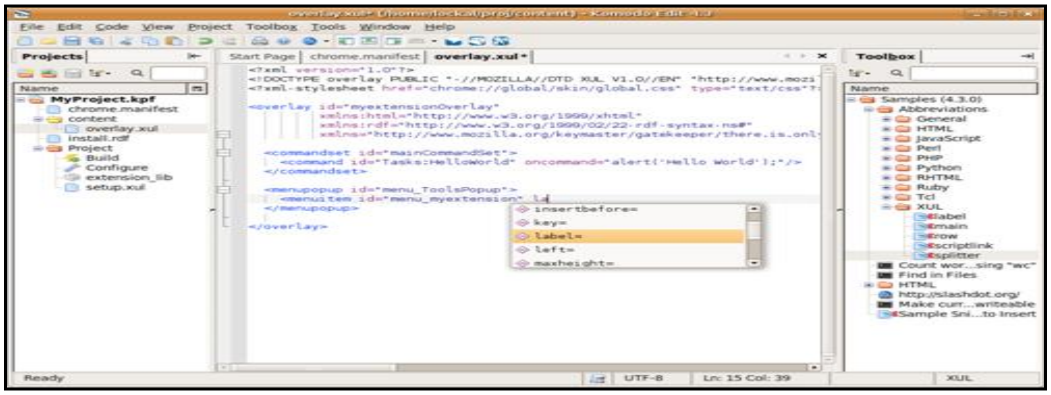
**Figure 12 Extreme Project Management Tool (XPMT)**

## 9. Extreme Programming Markup Language (XPML):

It is an open source tool that is available for free. It works on XML schemas. It includes sub versioning in source code **(Auer and Miller, 1996)**. It has release and iteration plans and metaphors included in the project.

## Limitations of Extreme Programming Markup Language:

- It has many functional defects.
- It cannot create backlogs.
- It is solely based on XML so it is not platform dependant.



```
<?xml version="1.0"?>
<!DOCTYPE overlay PUBLIC "-//MOZILLA//DTD XUL V1.0//EN" "http://www.mozi
<?xml-stylesheet href="chrome://global/skin/global.css" type="text/css"?

<overlay id="myextensionOverlay"
         xmlns:html="http://www.w3.org/1999/xhtml"
         xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.onl

  <commandset id="mainCommandSet">
    <command id="Tasks:HelloWorld" oncommand="alert('Hello World');"/>
  </commandset>

  <menupopup id="menu_ToolsPopup">
    <menuitem id="menu_myextension" la
  </menupopup>

</overlay>
```

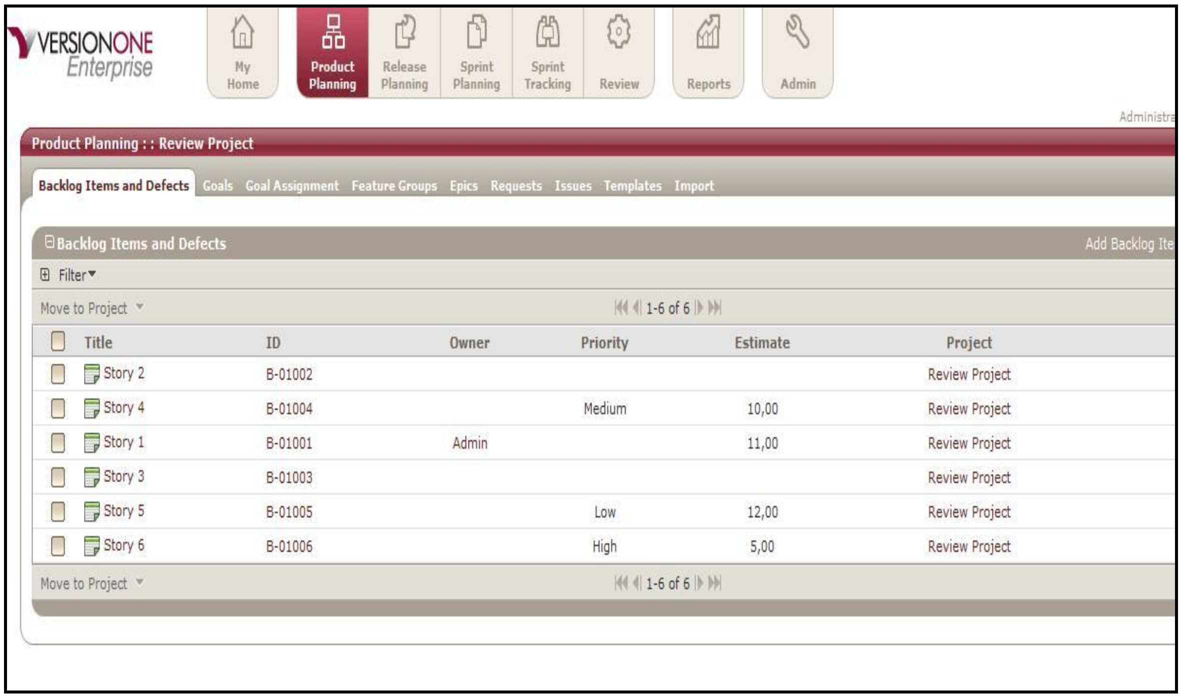
**Figure 13 Extreme Programming Markup Language**

## 10. Version One:

Version One is a extreme programming management tool that manages agile projects in the market. It also includes the scrum methodology. It also requires a significant financial investment.  It has different levels of planning and has maintenance of backlogs **(Cohn's, 2000)**. . The tool has a crystallized graphic user interface.

## Limitations of Version One:

- It is very expensive.
- The two modules are difficult to work and failure probable.



| Title | ID | Owner | Priority | Estimate | Project |
|---|---|---|---|---|---|
| Story 2 | B-01002 | | | | Review Project |
| Story 4 | B-01004 | | Medium | 10,00 | Review Project |
| Story 1 | B-01001 | Admin | | 11,00 | Review Project |
| Story 3 | B-01003 | | | | Review Project |
| Story 5 | B-01005 | | Low | 12,00 | Review Project |
| Story 6 | B-01006 | | High | 5,00 | Review Project |

**Figure 14 Version One**

## 11. Extreme Management Tool:

This tool provides project administration through Extreme Programming concepts. It is an outdated tool. It has not had a release in over two years and is no longer maintained.

## 8. Most Widely Used Extreme Programming Tools in the World

There are many tools available in the market today. The three that are mostly used are:

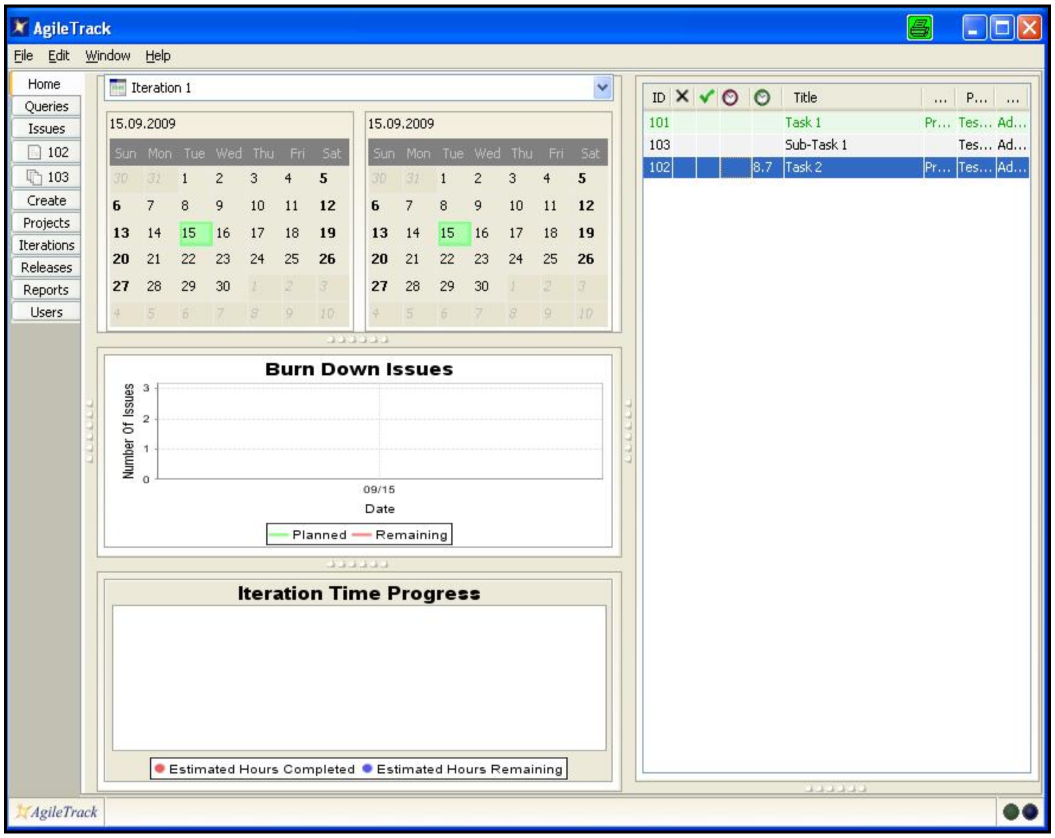
1. AgileTrack Tool
2. TaskJuggler Tool
3. PPTS - Project Planning & Tracking System

   The reasons for using these three tools are that these tools are more comprehensive regarding the latest functionalities of the Extreme programming technology. These tools are light threaded, easily expandable, and contains separate modules for all the activities in extreme programming.

1. **<u>Agile Track Tool:</u>**

   AgileTrack tool is a tool that supports the planning and management of Agile software projects. AgileTrack is a client rich application. It stores the data in a central or local managed database **(Cohn's, 2000)**. This tool enables the management, planning and control of task, release and iteration.

   Data is presented in the form of a list. A screenshot of the AgileTrack tool is given below



| ID | | | | | Title | ... | P... | ... |
|---|---|---|---|---|---|---|---|---|
| 101 | | | | | Task 1 | Pr... | Tes... | Ad... |
| 103 | | | | | Sub-Task 1 | | Tes... | Ad... |
| 102 | | | | 8.7 | Task 2 | Pr... | Tes... | Ad... |

**Figure 15 Interface of AgileTrack Tool**

## <u>Analysis of AgileTrack Tool:</u>

The features of AgileTrack Tool are:

- It is a very simple tool.
- It is based on Time Logging Concept. It breaks down the project and estimates the efforts based on how much time has been logged. .
- It manages task and issues.
- It is a client rich application. It is easily installed on the client site.

## Pros & Cons of the AgileTrack Tool:

- **Pros:**

**1)** It works on both Linux & Windows servers which are mostly used in cyber world.

**2)** It provides a web Interface.

**3)** It has defect management.

**4)** It has search functionality with quick query building function.

**5)** It has an easy & attractive Interface.

- **Cons:**

It does not produce reports.

It is a very basic tool and hence lacks advanced features.

It is not capable of handling large projects

2. **<u>TaskJuggler Tool:</u>**

TaskJuggler is a free and open source software management tool. It provides a complete range of project management tasks from the first idea to project completion **(Cohn's, 2000)**. It also provides an optimizing scheduler that will compute project deadlines, cost, revenue planning, resource assignments, risks and communication management.

The screenshot of the tool is:

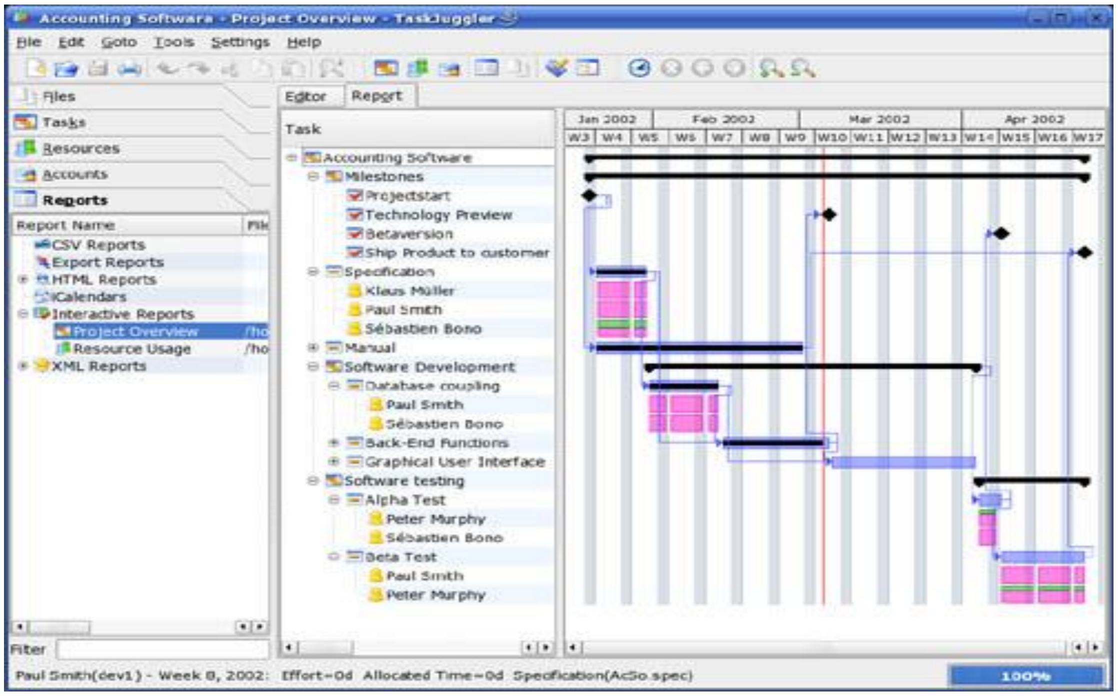


**Figure 16 Interface of Task Juggler Tool**

**<u>Analysis of TaskJuggler Tool:</u>**

The features of the TaskJuggler tools are:

- It is based on To-do list management.
- It manages project accounts, tasks & resources.
- It is integrated with text editor.
- It has automatic resource leveling.
- It has automatic tasks conflict resolution.
- It supports shift working scenario.
- It has flexible working hours.
- It has application for leave management.
- It accounts for all the costs incurred by the project including initial cost, finishing cost, resource based cost.
- It estimates profit/loss analysis.

**<u>Pros & Cons of the TaskJuggler Tool:</u>**

- **<u>Pros:</u>**

1) Scalable. Projects can be combined to larger projects.
2) It supports central resource allocation database.
3) It presents output in HTML reports.
4) It has a built-in web server.
5) It has dashboard support for status reporting and project reporting.

6) It has CSV data export for data exchange.

- **<u>Cons: (please number these too)</u>**

  Advanced Web Server Requires.

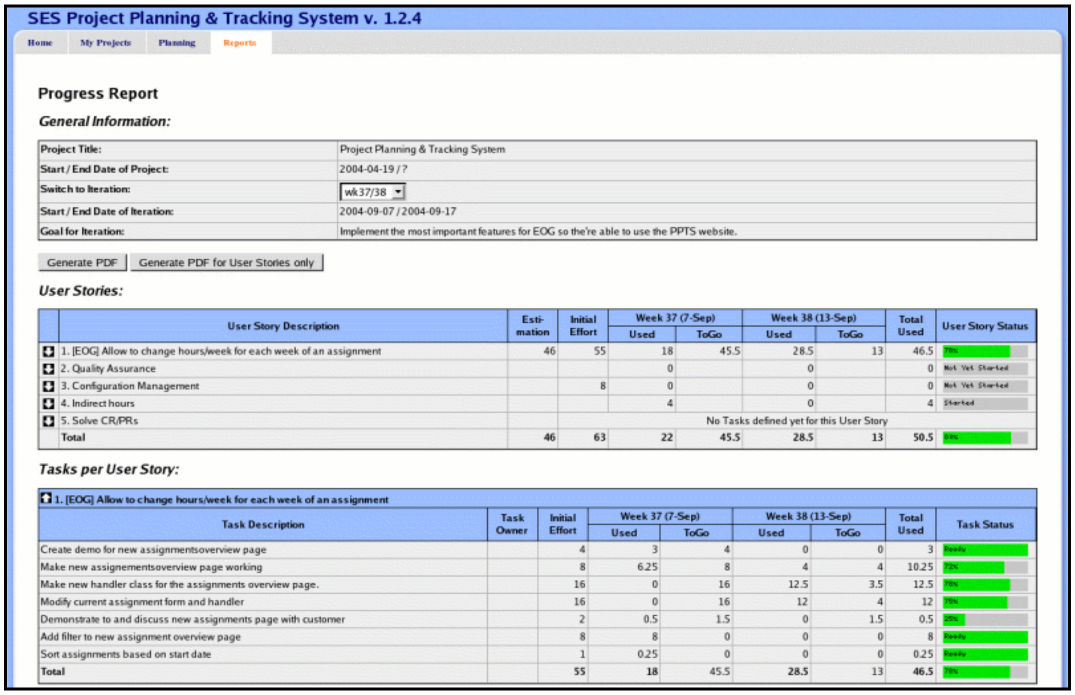
  High Cost

  Once a Webpage is closed, the server cannot retrieve the TaskJuggler data again.

## 3. <u>PPTS - Project Planning & Tracking System:</u>

PPTS is a web based environment team who develop the software according to the agile methodologies **(Cohn's, 2000)**.

The screen shot of the tool is:



SES Project Planning & Tracking System v. 1.2.4
Home
My Projects
Planning
Reports
Generate PDF
Generate PDF for User Stories only

Progress Report

*General Information:*

| | |
|---|---|
| Project Title: | Project Planning & Tracking System |
| Start / End Date of Project: | 2004-04-19 / ? |
| Switch to Iteration: | wk37/38 |
| Start / End Date of Iteration: | 2004-09-07 / 2004-09-17 |
| Goal for Iteration: | Implement the most important features for EOG so the're able to use the PPTS website. |

*User Stories:*

| User Story Description | Esti-mation | Initial Effort | Week 37 (7-Sep) | | Week 38 (13-Sep) | | Total Used | User Story Status |
|---|---|---|---|---|---|---|---|---|
| | | | Used | ToGo | Used | ToGo | | |
| 1. [EOG] Allow to change hours/week for each week of an assignment | 46 | 55 | 18 | 45.5 | 28.5 | 13 | 46.5 | 70% |
| 2. Quality Assurance | | | 0 | | 0 | | 0 | Not Yet Started |
| 3. Configuration Management | | 8 | 0 | | 0 | | 0 | Not Yet Started |
| 4. Indirect hours | | | 4 | | 0 | | 4 | Started |
| 5. Solve CR/PRs | | No Tasks defined yet for this User Story | | | | | | |
| Total | 46 | 63 | 22 | 45.5 | 28.5 | 13 | 50.5 | [illegible] |

*Tasks per User Story:*

1. [EOG] Allow to change hours/week for each week of an assignment

| Task Description | Task Owner | Initial Effort | Week 37 (7-Sep) | | Week 38 (13-Sep) | | Total Used | Task Status |
|---|---|---|---|---|---|---|---|---|
| | | | Used | ToGo | Used | ToGo | | |
| Create demo for new assignmentsoverview page | | 4 | 3 | 4 | 0 | 0 | 3 | Ready |
| Make new assignementsoverview page working | | 8 | 6.25 | 8 | 4 | 4 | 10.25 | 72% |
| Make new handler class for the assignments overview page. | | 16 | 0 | 16 | 12.5 | 3.5 | 12.5 | 78% |
| Modify current assignment form and handler | | 16 | 0 | 16 | 12 | 4 | 12 | 75% |
| Demonstrate to and discuss new assignments page with customer | | 2 | 0.5 | 1.5 | 0 | 1.5 | 0.5 | 25% |
| Add filter to new assignment overview page | | 8 | 8 | 0 | 0 | 0 | 8 | Ready |
| Sort assignments based on start date | | 1 | 0.25 | 0 | 0 | 0 | 0.25 | Ready |
| Total | | 55 | 18 | 45.5 | 28.5 | 13 | 46.5 | 78% |

**Figure 17 Interface of PPTS**

**Analysis of PPTS Tool:**

The features of PPTS tools are:

- It manages administration, iteration and resource attributes.
- It presents output in data backlogs.
- It is based on the work breakdown structure.
- It uses Estimated/Spent Metrics to Estimate Cost.
- It prepares progress charts.
- It works on resource allocation.

**Pros & Cons of the PPTS Tool:**

- **Pros:**
  1) It has the ability for customizing menu and languages.
  2) It is an Open Source Tool
  3) It grants information access based on roles.
  4) It interfaces with the PR/CR tools.
- **Cons: (number them please)**

  The web server is of high configuration.

  The network congestion needs to be taken care of.

## 9. Comparison of Three Main Tools

**Table 3 Comparison of Three Main Tools**

| **Terms** | **AgileTrack** | **TaskJuggler** | **PPTS** |
|---|---|---|---|
| **Management** | Task & Issue Management | To-Do List Management | Project Management, Iteration & Resource Attribute Management |
| **Capability Maturity Model (CMM)** | CMMI | CMM level 3 | CMM level 2 |
| **Data Presentation** | List Format | Dynamic & Interactive Report | Prioritizable Product Backlog |
| **Scheduling** | Resource Allocation Based On Time Logging | Automatic Resource Leveling | Resource Allocation Based On Job |
| **Operating System Support** | Linux & Windows | Linux, Unix, Windows, MacOS and several OS. | Windows & MacOS |

**Table 3 (continued)**

| | | | |
|---|---|---|---|
| **Working Methodology** | Breaks Project in to Well Defined Blocks & Interfaces. | Breaks Project In Unlimited Scenarios. Works On What-If Basis. | Works On Breakdown Structure |
| **Scaling** | Scaling Not Possible. Very Basic Tool. | Scaling Possible. Projects can be combined to large projects. | Scaling Possible. |
| **Complexity** | Very Basic Tool. Lacks Advanced Features Like Reporting & Collaboration | Highly Advanced. HTML Report, CSV Data Export, iCalendar Export For Data Exchange. | Advanced. Customization Of Menus & Language Possible. Prioritizable Backlogs Possible. |
| **Role Management In Team** | Access To Information Is Equally Divided As AgileTrack Is For Small Projects | Access To Information Based On Constraints Of The Team | Fine Access To Information Based On Overall In Project. |

**Table 3 (continued)**

| | | | |
|---|---|---|---|
| **Installation** | Very Simple Installation. Documentation Available. | Simple Installation. Detailed Reference Manual. | Simple Installation Of Web Server. |
| **Server** | Server Available | Built-In Web Server Available For Time Sheet Management | Web Server Available |
| **Cost** | $39/User | $65/User | $47/User |

## 10. Conclusion

**Table 4 Strengths and Weaknesses of Tools**

| **TOOL** | **STRENGTHS** | **WEAKNESSES** | **COMMENTS** |
|---|---|---|---|
| **Apache Ant** | Integrate into all IDE's<br><br>Ant is java based and written in xml. | Very complex structure and the scripts are very large | Ant is the de-facto standard build tool for Java projects |
| **Project Engine** | Available for Free<br><br>Provides features like Gantt charts, time reporting, scheduling | Lacks enhancement | Simple design, quick installation and simple configuration. |
| **Rally Agile** | Provides integration and customization features.<br><br>Developers focus on the important issues. | Customized reports can't be generated. | Enterprise proven agility tool.<br><br>Low cost. |

**Table 4 (continued)**

| | | | |
|---|---|---|---|
| **Target Process** | Automates all the tasks associated with the project<br><br>Development process, workflows, plan iterations, releases can be customizable | Require an IIS application server | Integrates testing process with the tracking process and project planning. |
| **Track IT** | Web based project tracking tool<br><br>Open Source<br><br>Integrate with CVS | | Built in support for programming |
| **Extreme Planner** | Web based tool<br><br>Support Agile methods like Scrum. | No reports customization | Agile project management tool for the team that operate from different locations |

**Table 4 (continued)**

| | | | |
|---|---|---|---|
| **XP Web** | Web based tool.<br><br>Supports XML Export functionality | Doesn't support releases and iterations | Supports on windows and Linux platforms |
| **XPMT** | Open Source tool<br><br>Supports features like planning, iteration tracking, release tracking, time planning. | Very Complex structure and slow processing | |
| **XPML** | Open source and available for free<br><br>Supports release and iteration plans | Many functional defects<br><br>Can't create backlogs | Purely XML based and platform independent |
| **VersionOne** | Supports Scrum Methodology.<br><br>Supports different levels of planning. | Very costly | Tool has a crystallize graphic UI. |

**Table 4 (continued)**

| | | | |
|---|---|---|---|
| **XPlanner** | | | |
| **PPTS** | Web based Environment<br><br>Open source tool<br><br>Based on Work Breakdown structure<br><br>Interface with PR/CR tools | High Configuration web server is required.<br><br>Network congestion needs to be taken care. | Supports multiple languages (I18N) |
| **Extreme Management Tool** | Project administration through Extreme programming concepts | Out Dated tool. | No releases in the last two years and no longer maintained |

**Table 4 (continued)**

| | | | |
|---|---|---|---|
| **Agile Track** | Very Simple tool<br><br>Manages Tasks and Issues<br><br>Provides Web Interface | Do not produce reports<br><br>Very basic tool and lacks advanced features<br><br>Not used for large projects. | AgileTrack is client rich application. Supports both windows and linux environments |
| **TaskJuggler** | Free and Open Source<br><br>Manages project account, tasks and resources.<br><br>Estimates Profit and loss analysis of project<br><br>Supports HTML reporting. | More expensive | Provides complete range of project management tasks from first idea to completion. |

Based on the research of all the XP tools mentioned above that are available in the industry, there is no single tool available in the market as of today that will help though out all the practices of XP. Each tool covers only a specific area and has strengths and weaknesses. **AgileTrack** is a very basic tool and cannot manage big projects. **TaskJuggler** is a very advanced tool. It has a scalability option which is more advanced than PPTS.  Its presentation is also in the form of reports. However the interface is complex. **PPTS** is an open source tool. It is known for the customization of its menus. The metrics used to estimate cost is accurate.

Evaluating these tools with the XP practices mentioned at the beginning of this report, it is found that every tool supports either partially or fully support throughout the XP practices. Some of these tools need a highly configured server to take care of all the functionalities. Agile Track tool provides support for Risk Management, Scope Management, Time Management and Quality Management but lacks advanced features and not used for large projects. Task Juggler tool supports all the features but it is an expensive tool. PPTS is one of the best open source tools that support all the features.  However this tool requires a highly configured server and needs to take care of the network congestion.  The table below gives the list of XP practices supported by these tools.

Table 5 Evaluate Tools with XP Practices

| XP Practices | Agile Track | Task Juggler | PPTS | Project Engine | Rally Agile | Target Process |
|---|---|---|---|---|---|---|
| Risk Management | ✓ | ✓ | ✓ | | | |
| Integration Management | | ✓ | ✓ | ✓ | ✓ | ✓ |
| Scope Management and Time Management | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Human Resources Management | | ✓ | ✓ | ✓ | | |
| Quality Management | ✓ | ✓ | ✓ | | ✓ | ✓ |
| Communications Management | | ✓ | ✓ | ✓ | | ✓ |

## 11. References

- Succeeding With Agile By Cohn's, Mike (Edition - 2000).
- Planning Extreme Programming by Beck, Kent and Fowler, Martin (Edition - 1998).
- Extreme Programming Applied: Playing To Win By Auer, Ken and Miller, Roy (Edition - 1996).
- Extreme Programming Explored By Wake, William C (Edition - 1997).
- White Papers on Extreme Programming by Menloinnovations (Edition - 2004).
- White Paper On Extreme Applied To IP Development By Maojet Corp (Edition - 2006).
- Beck. Kent, 2000. Extreme Programming Explained. ISBN 0-201-61641-6, Addison- Wesley.
- Williams, L., Kessler, R. 2001. Experimenting with Industry's Pair-Programming Model in the Computer Science Classroom, Journal on Computer Science Education.
- Jacobson, I. 1992, Object-Oriented Software Engineering: A Use Case Driven Approach, ACM Press.
- Jeffries, R., Anderson, A., Hendrickson, C. 2001, Extreme Programming Installed. Reading, Massachusetts, Addison-Wesley.
- Fowler, Martin 2003. Refactoring: Improving the Design of Existing Code, 2003, DOI= http://www.martinfowler.com/books.html/refactoring
- ManLui, Kim Chan, Keith C.C. 2003. When does a pair outperforms two individuals? in Proc. Fourth International Conference on Extreme Programming and Agile Processes in Software Engineering,, Genova, Italy, DOI= http://www.xp2003.org/slides/15.pdf