

Simple Progress Reporting

Neil Ashworth

Synop Pty Ltd
PO Box 1024
Artarmon NSW 1570
Australia
+61 2 9411 8744
neil@synop.com

Peter Bailey

Synop Pty Ltd
PO Box 1024
Artarmon NSW 1570
Australia
+61 2 9411 8744
peter@synop.com

Nathan Wallace

Synop Pty Ltd
PO Box 1024
Artarmon NSW 1570
Australia
+61 2 9411 8744
nathan@synop.com

ABSTRACT

This paper present briefly examines the ongoing divide between management and IT, and examines some practical ways of bridging the communication gap by reporting project progress in business friendly manner..

It looks at methods for estimating the size of a project using standard techniques such as guessing, function point analysis, backward scheduling and others, converting these to standard units of work in an XP project, and then communicating progress to the development team and to management.

Keywords

Management, estimating, measuring, reporting, risk,

1 THE GAP

The relationship between business and IT has long been one of wary watchfulness and distrust. The extravagant success of outsourcing firms late last century would not have happened if business owners believed their own IT departments could have delivered the promises we have been making for the last forty years about improved productivity, lower costs, and increased competitiveness. Similarly, what software development company is not familiar with the smirk of disbelief that flickers across the face of every prospect when the quotes are presented.

So after Cobol, relational databases, 4GLs, client server, e-business and ERPs have successively failed to deliver even a fraction of their promised benefits, why are we surprised with the disbelief and cynicism that greets our plans to introduce a new development method with the dodgy name of eXtreme Programming.

2 BUILDING TRUST

What the business owner wants is a reduction in risk,[6] which will help reduce the hidden problem, fear[3].

The approach we need to take on this is multi pronged approach to reducing risk. To do that we need to know some vital information:

1. What is the purpose of the project
2. How do we reliably estimate duration and cost
3. How do we ensure that progress is reported in a comprehensible and simple manner

3 ESTIMATING

Libraries have been written on human attempts to pre-

dict the future, and in our little part of the universe Rob Thomsett has covered the topic better than most.

Do it at least three ways, for example:

1. Guessing
2. Guestimating
3. Function point counting
4. Backward scheduling
5. Resource loading

Leave estimating as late as possible.

Estimate from the bottom up, whatever the bottom is at this point in time.

And then you can convert your estimate into “stories” or “features”. For this you need some historical data on the performance of your team, in your organization, using this technology. If you haven’t a reasonable rule of thumb is 5 days per story per person. Of course this is going to be subject to diminishing returns as the size of the project team grows.

4 PROGRESS REPORTS

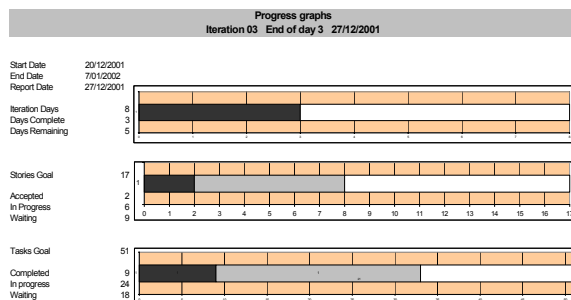
The progress reporting presented in Planning Extreme Programming is simple, sensible and suitable for the team. The question then is how to present progress of the project. We usually employ three levels of report:

1. Iteration progress measuring time against progress in stories and tasks
2. Release progress, measuring time against stories completed.
3. Project progress measuring stories completed and money spent against the original budget.

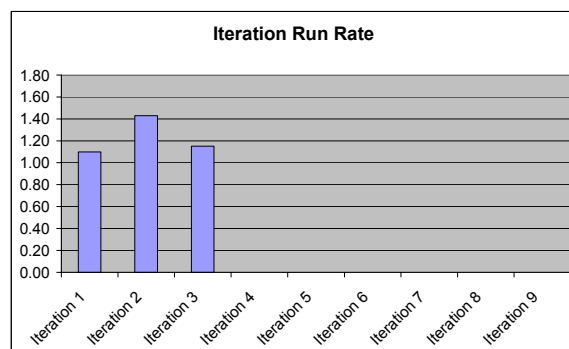
5 ITERATION PROGRESS

Iteration reporting is the base level of progress reporting. These reports need to be comprehensible to the project team, and also be usable as detail evidence in management reporting.

The purpose is not only to show progress, but also to show the distance from the goal.



The following graph also shows the progress being achieved in each iteration, and whether gains in productivity are being achieved over like sized iterations.



6 RELEASE PROGRESS

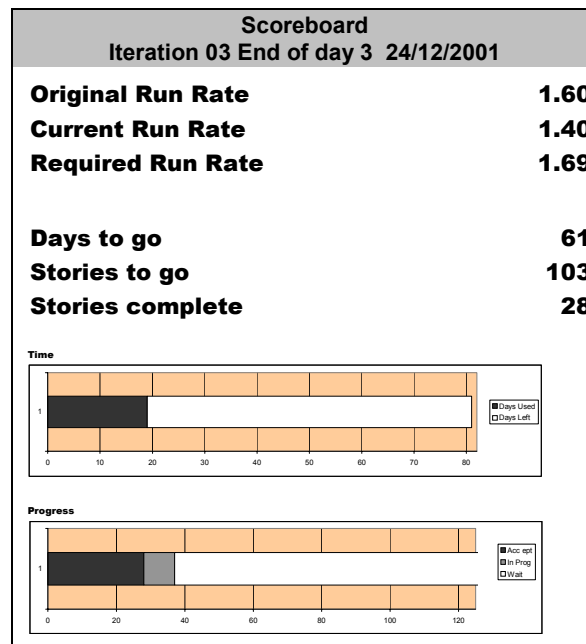
Release progress is best measured by what we call the “Run Rate”. This is a simple measure of stories per day by the team, but also includes the message clear to all cricket fans that there is an objective to be achieved, and how close to achieving it we are.

This is combined with a very simple progress gauge. Progress is divided into Waiting (not yet started), In Progress (work has started) and Accepted (customer or QA has accepted this story). We do not use the word “Complete” as it has too many different meaning to different participants. (See figure on the top right of the page.)

7 FINANCIAL PROGRESS

Measuring financial progress has not been touched on in any of the standard XP books, probably because the genesis of XP was in the programming community, historically unconcerned with any thoughts of money except that coming in their direction.

The use of Earned Value Management that relates resources to schedules to cost to requirements should lend itself to an easy adaption to the XP process with iterations and release delivering specific value. By the time of the conference I plan to have some prototypes of this analysis.



8 AUTOMATION AND TOOLS

I have seen a number of prototypes, designs, simple working models and some quite complex computer based systems for measuring progress.

In each case the decision needs to be made as to whether the tools will make the development process more streamlined and automatic, will they INCREASE the communication within the team, and are they flexible to cope with the “ecosystem” metaphor of XP.

9 SUMMARY

Measuring progress in an XP project is vital, not just for the team members, but for all project stakeholders. It is important not just to make progress, but to demonstrate that progress is being made, and value is being earned by the organization in return for its financial investment.

REFERENCES

1. Robert J. Benson. Business and IT: A Gap Analysis, *Business IT Strategies Advisory Service* Vol 4 No 8
2. Neal Whitten. Managing Software Development Projects, John Wiley & Sons.
3. David Maister. *Managing the Professional Service Firm*. The Free Press.
4. Chris Gane. *Rapid System Development*. Prentice Hall
5. Kent Beck, Martin Fowler *Planning Extreme Programming*,
6. Nathaniel Talbot, Roy W Miller, *Selling XP to People Who Buy*