

Verification Leadership Seminar

- Predictability
- 17 January 2006

Agenda

- Stating the Problem
- Employed Solutions and their Drawbacks
- Exploration of new Solutions
- Summary

How Can We Predict Verification Progress?

Most of us encountered this problem in two stages of the project:

- In the Work Plan stage before the project starts we need to supply a time estimate on the verification stage, or if I may rephrase, we often need to estimate the amount of skill and risk involved in going to Tape-out at a specific date!
- During the Verification stage we need a way to set goals and track the progress of the verification.

Employed Solutions and their Drawbacks

- Software like: tracking bug quantity and frequency.
=> Very poor prediction results and unreliable results but quite easy to present to management.
- Tracking the code coverage progress.
=> Reliable results, but very heavy on resources and poor at predicting the verification complexity, progress.
- Tracking the functional coverage progress.
If done properly reliable results, but still hard to predict the progress due to interaction with other teams and unknown design complexity.

Exploration of new Solutions

- Building a statistical model that takes most of the parameters and gives an estimate on the expected progress curve.
- Doing the coverage tracking per block but checking the estimate on the overall project not a specific block.
- Combining different methods in different stages:
For initial resources estimate using previous experience data.
For bring-up stage using bug quantity and frequency method.
Combining functional coverage with code coverage for final stages and measuring each blocks progress vs the overall progress.

Summary

- The ability to predict verification is still a magic more than a science since therefore mistakes will be made and we need to take it into account in work plans.
- The problem to predict accurately the verification progress is closely related to our inability to measure verification quality.
- Code coverage and functional coverage seems to be inescapable for good predictability.
- Predicting a progress of a single block seems unrealistic due to the number of unknown factors, yet averaging between blocks can help.

Verification Leadership Seminar

- Predictability
- 17 January 2006

Agenda

- Stating the Problem
- Employed Solutions and their Drawbacks
- Exploration of new Solutions
- Summary

How Can We Predict Verification Progress?

Most of us encountered this problem in two stages of the project:

- In the Work Plan stage before the project starts we need to supply a time estimate on the verification stage, or if I may rephrase, we often need to estimate the amount of skill and risk involved in going to Tape-out at a specific date!
- During the Verification stage we need a way to set goals and track the progress of the verification.

Corrigent Confidential

Trying to estimate the resources needed for a given project before it starts is a complicated problem. A great deal of data is missing in this stage of the project. In this stage of the project it is very likely that the only information available to us is the major requirements and the end date deadline.

The majority of the projects do an estimate here based on “gut feeling” of the manager, some projects have tried to rely on previous projects experience. Although it sounds fair, there are still problems involved and a great deal of inaccuracy;

- Projects have different complexity and different goals.

The teams that work on the project are usually not the same teams causing inaccuracy in estimating the work quality – this is even worse for the verification stage as verification quality is currently unmeasurable!

During the verification stage we need to set goals for individual team members, realistic goals yet ones that will bring the end result at the expected time, basically this is a standard managerial job but nothing here is standard! The progress of the tasks does not depend only on the team member but also on the other teams that we interact with (design, software system etc...) and on computing resources.

The first stage of verification, the bring-up of the design, is the hardest to predict – having major interaction between the designer and verifier. Later comes a stage of sharp progress curve having a behavior similar to exponential polynomial, what are the parameters and grade of the polynomial is unknown.

The last stage is also hard - in this stage we need to get to 100% coverage, the complexity of getting to this goal depends on the following parameters:

- Block complexity – gate count / architecture / amount of shared resources
- Work quality – how good is the design and verification
- Computing resources

Employed Solutions and their Drawbacks

- Software like: tracking bug quantity and frequency.
=> Very poor prediction results and unreliable results but quite easy to present to management.
- Tracking the code coverage progress.
=> Reliable results, but very heavy on resources and poor at predicting the verification complexity, progress.
- Tracking the functional coverage progress.
If done properly reliable results, but still hard to predict the progress due to interaction with other teams and unknown design complexity.

Exploration of new Solutions

- Building a statistical model that takes most of the parameters and gives an estimate on the expected progress curve.
- Doing the coverage tracking per block but checking the estimate on the overall project not a specific block.
- Combining different methods in different stages:
For initial resources estimate using previous experience data.
For bring-up stage using bug quantity and frequency method.
Combining functional coverage with code coverage for final stages and measuring each blocks progress vs the overall progress.

Summary

- The ability to predict verification is still a magic more than a science since therefore mistakes will be made and we need to take it into account in work plans.
- The problem to predict accurately the verification progress is closely related to our inability to measure verification quality.
- Code coverage and functional coverage seems to be inescapable for good predictability.
- Predicting a progress of a single block seems unrealistic due to the number of unknown factors, yet averaging between blocks can help.