

Rapid Test Planning

Quality quickly

Joy Shafer

Test Lead

Quardev Laboratories -- Seattle

joys@quardev.com

SASQAG -- July 14, 2004



What is Rapid Test Planning

and why would you want to do it?

What

- Fast
- Informal
- Focuses on getting the testing done
- No formal documents required

Why

- Use when time is critical
- Use when formal documentation is not needed
- The techniques are universal



When time and resources are short, planning is crucial

Why Plan your Testing?

“Planning testing” is not the same as “writing a test plan”

- Planning helps you focus your testing on important areas
- Planning helps you define scope
- Planning helps ensure you don't forget something critical
- Planning helps you manage process, people and time
- Planning provokes conversations that will allow you to do better testing



The Trade-Off

“Time spent planning is time spent *not* testing”



Rule of Thumb: 10 to 15% of project time should be spent on planning

Mitigating the Trade-Off

- Start planning right away
- Start testing right away
- Watch your time carefully



Mitigating the Trade-Off

Start planning right away

- Don't wait for the build to arrive
- Use whatever documentation is available: requirements, developer docs, user docs, bug reports, features lists
- Use your own knowledge of the project and technologies
- Investigate: do research, contact people in the know
 - Developers, PM, tech support, other testers, technical writers, marketing



Mitigating the Trade-Off

Start testing right away

- Don't wait for the plan to be "finished"
- As soon as the software arrives, have testers jump on it!
 - Go through the menus
 - Do "user" actions
 - Regress old bugs
 - Use "exploratory testing" techniques
 - Follow industry "best practices"



Mitigating the Trade-Off

Watch your time carefully

- Beware of “black hole” activities
- Don't be afraid abandon a course if it is taking too long
- Enlist tester help, but set expectations for how much time they should spend



“Good Enough” Test Planning

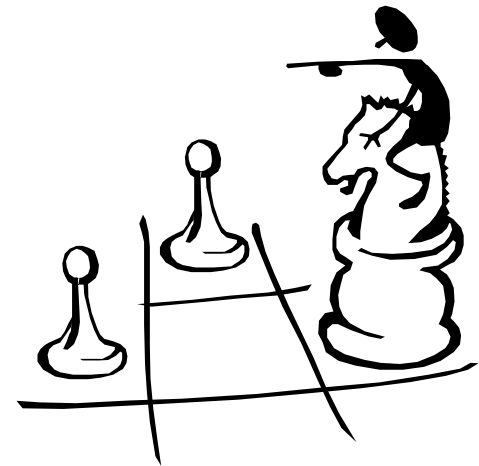
How do you know when you’ve done enough planning?

1. Sufficient Benefits
2. No critical problems
3. The benefits outweigh the problems
4. All things being equal, further planning is more harmful than helpful

It’s a matter of economics

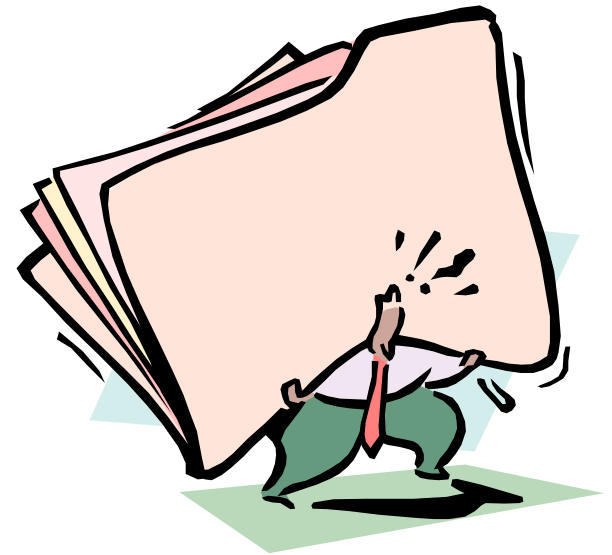
Rapid Test Planning Activities

- Gather information
- Communicate!
- Identify high risk areas
- Create (only) needed documentation
- Test smart
- Communicate!
- Revisit the plan
- Do a post-mortem



Information Gathering Results

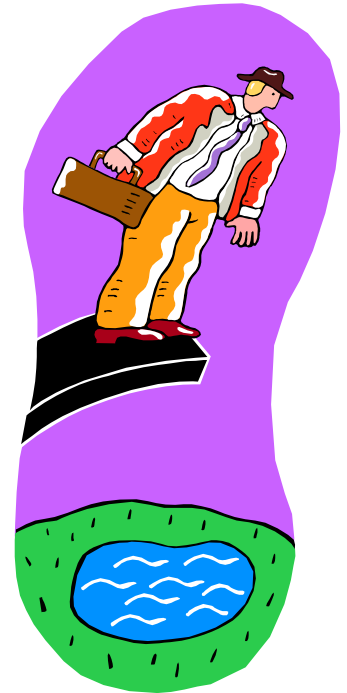
- Stacks of documents
- Lists of references / links
- Lists of questions, concerns and, hopefully, answers
- List of contacts
- Relationships fostered



Identify High-Risk Areas

Risk = Likelihood + Impact

What's the likelihood of something bad happening, and how bad would it be if it did?



High Risk Software Areas

- Areas that are new
- Areas that have recently changed
- Areas that have been buggy in the past
- Areas of complexity
- Areas of new technology
- Areas of low testability
- Mainstream functionality
- Critical functionality

High Risk Project Factors

- Special resources that have yet to be acquired / set up
- Long distance relationships
- Logistical concerns



Risk Analysis Results

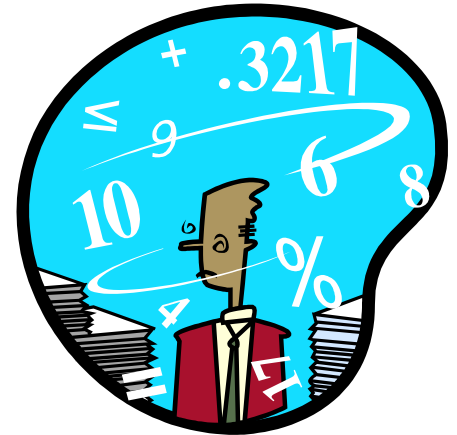
- List of high-risk product areas
- Ideas for how to test them
- List of high-risk project factors
- Ideas for how to mitigate them
- Additional questions, concerns and, hopefully, answers



What Docs Are Critical?

Decide what documentation is really critical

- Docs for testers
 - What documents will best help manage the testing process?
 - What documents will best help track coverage?
- Docs for others
 - Can these be put off until after-the-fact?



Slow Docs

- Formal test cases and scripts
- Test automation
- Long formal documents



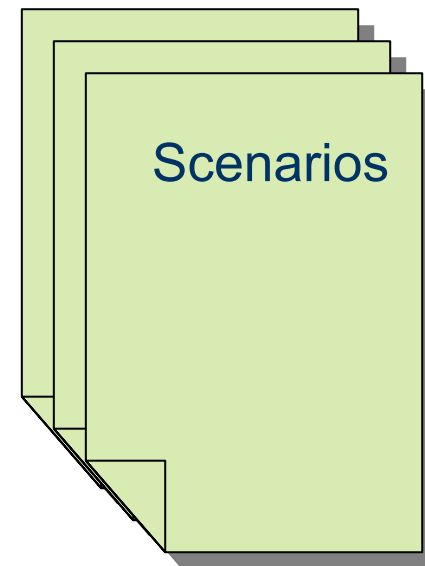
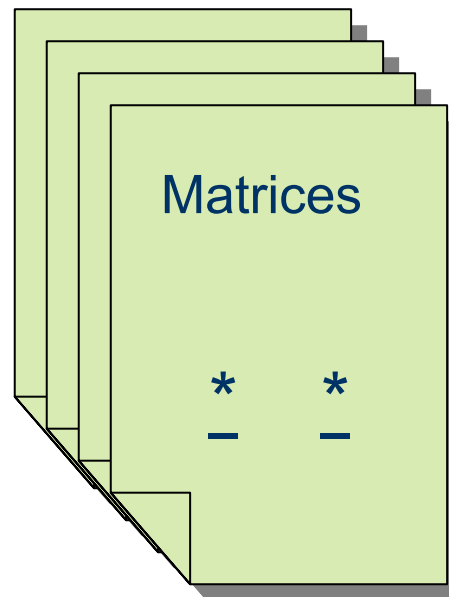
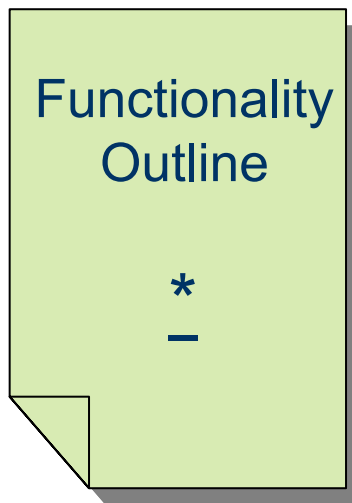
Fast Docs

- Scenarios
- Matrices
- High level functionality outlines
- Quick and dirty automation
- Tests from other projects
- Bug reports from existing or previous bugs



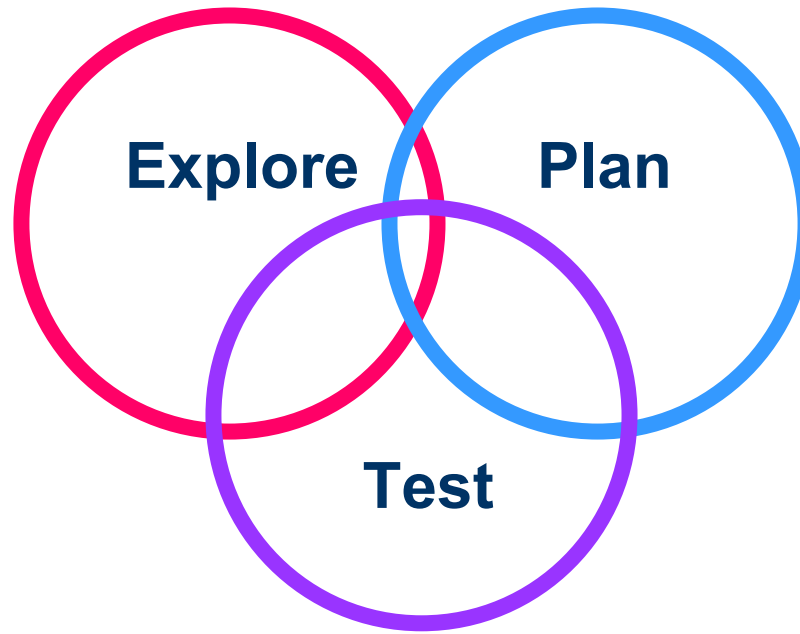
Documentation Results

- Appropriate documentation for efficient and effective testing



Exploratory Testing

Exploratory testing is the simultaneous process of exploring an application, looking for bugs, and developing a test plan.

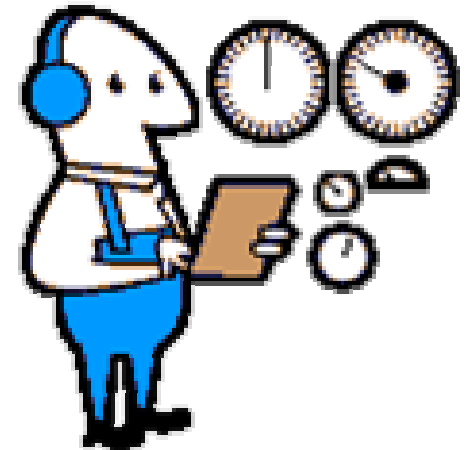


Test Smart

Test expected inputs first...

- Test ideal path actions with expected data
- Boundary test reasonable inputs
- Test expected invalid inputs
- Do compatibility test with expected configurations

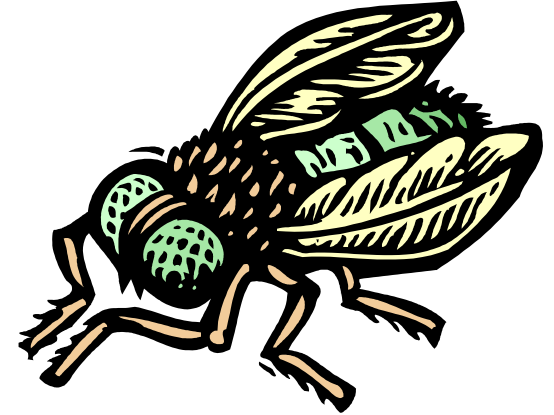
...crazy inputs last



Pay attention to high risk areas!

Testing Results

- Bug reports
- Updated bug reports (from regression testing)
- Status reports
- Coverage knowledge
- Knowledge of what is working and what is not working



Revisit the Plan

- Reset priorities
- Balance resources
- Communicate!



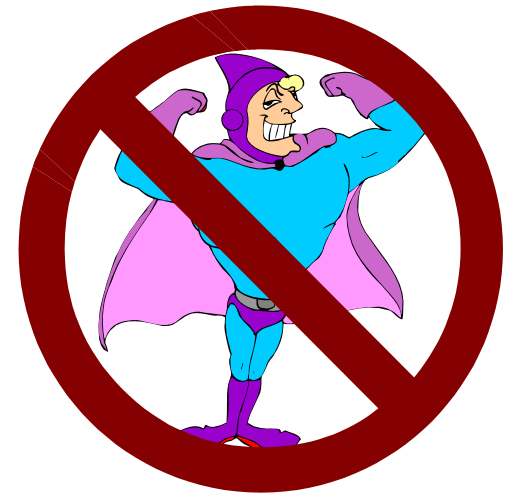
“Planning is essential. Plans are useless.”

--Dwight Eisenhower

Project Postmortem

Resist heroics

- Point out coverage holes - everything that was not tested due to lack of time
- Don't lie
- Point out extreme work hours, unreasonable resource usage, etc.



Postmortem Results

- Education of others about the testing process
- Better future test planning
- Better future testing
- Better working relationships



Drawbacks of RTP

Some projects require more in depth test planning

- It may be difficult to use if team is inexperienced and/or software is very complex
- It's good for one-shot testing. It may not be efficient in the long run
- There is no opportunity to educate others about the testing process

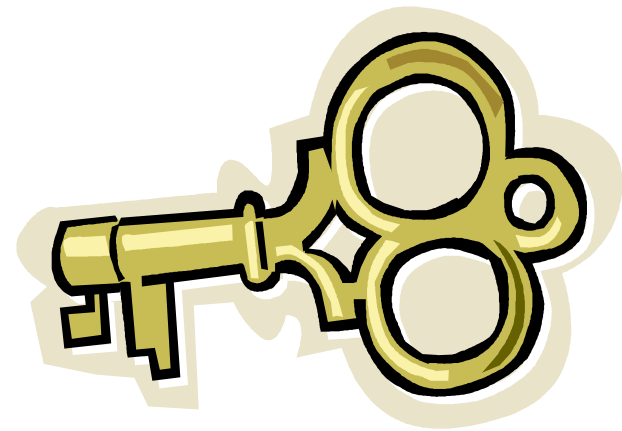
What RTP is Not

- Slipshod
- Formal
- Rigorous
- An excuse for shortening testing time



Key Ideas

- Effective test planning does not always involve creating a “test plan”
- Test planning can be done quickly
- Planning is particularly important for time-critical projects



Sources / More info

James Bach

“The Challenge of Good Enough Software”

<http://www.satisfice.com/articles/gooden2.pdf>

James Bach

“Quick and Dirty Test Outlining”

<http://www.satisfice.com/>

James Bach

“Exploratory Testing”

<http://www.satisfice.com/articles/et-article.pdf>

Cem Kaner

“Concise Test Planning”

(Tutorial session) *BOSCON 99* (American Society for Quality, Boston, annual conference),
Newton, MA, March, 1999

Derek Sisson

Testing Without a Formal Test Plan

http://www.philosophie.com/testing/without_testplans.html

Questions?

