Focused Iterative Testing: A Test Automation Case Study
Mechelle Gittens, Pramod Gupta, David Godwin, Hebert Pereyra, Jeff Riihimaki

DB2 Data Management Software
Problem Definition: Find Timing Related Defects

- Server software (e.g. DB2) is a multithreaded, multiprocess system with a large number of concurrent users
- Server software uses logging, locking/latching
- This causes Timing Related Defects (TRD)
- Difficult to reproduce customer problems
Different Approaches to Address trigger TRDs

- Vary CPU load by external programs
- Instrument code to put in random sleeps
- Change priorities of processes
- Iteratively execute commands with background database workload (Most Successful approach)
What is Focused Iterative Testing?

- Find a focus area which has timing related defects e.g. crash recovery

- Identify what finite number of commands are to be tested e.g. restart database command after a crash

- Write a tool to automate the whole test case and control the background workload

- The tool is run repeatedly till the target number of successful iterations is reached frequently. Track the number of successful iterations as a function of time.
Why do we use FIT?

- Improve reliability of mission critical components in DB2 e.g. crash recovery, monitor, FCM etc.
- FIT gives us an objective metric of quality which is very useful to management
- Easier to find and easier to debug Timing Related Defects
- Each FIT setup has different environment e.g. database configuration parameters
How do we write a FIT tool?

- Input from APAR (Customer Defects) Causal Analysis
- Input from Customer Critical Situations
- Input from developers (weekly meeting with developers)
- Implement FIT Tools
- Ensure tool is easy to use so more testers use it
Positive Feedback Loop

- Early implementation of Agile methodologies:
  - Weekly meeting with developers
  - 1. Meet developers to discuss previous week's defects and get suggestions for enhancing FIT tool
  - 2. Enhance FIT tool
  - 3. Run tool and find more defects
  - 4. Goto step1
Case Study: Crash recovery for a database server

- Stringent Requirement: A database server must keep the database in consistent state regardless of external events like power failures etc.

- Crash Recovery: When database is restarted after a outage then it must correctly replay the transaction logs and ensure that the database is in a consistent state.

- Crash recovery failure seriously affects customer's business and is a customer critical situation.

- Hence, it is very important to test crash recovery.
FIT tool for crash recovery

Timing Defects due to logging and replaying of logs

FIT crash recovery tool:

• Run clients issuing SQL and DB2 commands for a database
• Deliberately crash the system by killing DB2 processes
• Restart the database so server plays transaction logs.
• Repeat
• If the FIT tool finds a problem, it sends e-mail to tester.
• FIT tools for crash recovery are run on AIX and Windows
FIT tool for Fast Communication Manager Testing and for Monitor Testing

Timing defects due to buffers and channels between nodes.

Fast Communication Manager FIT tool:
• Similar to tool for crash recovery testing but we only kill selected nodes in a multi-node database system.
• Ensure communication between still running nodes is OK.
• Failure of subset of nodes should not bring other nodes down.
FIT tool for Monitor Testing

Timing defects since code has lot of locking of resources

Monitor FIT tool:
• run finite set of monitor commands with background workload
• Snapshot family of commands give current picture
• Event monitor commands give historical picture
• We do not kill any nodes
To FIT or not to FIT

FIT is an approach for System Testing

- It can be used to detect major defects like traps or hangs in code with timing related defects

- It is not applicable to:
  - Code which does not have timing related defects
  - Function testing
  - Unit testing
Investment in FIT

- At beginning of test cycle:
  - Extra tooling effort

- During test cycle:
  - Weekly meetings with development
  - Enhancing FIT tool
Return on Investment in FIT

- Larger number of timing related defects found during test cycle. So customer gets more reliable product.
- Problems found earlier in cycle
- Mean number of successful iterations gives objective measure of component quality
- To ship or not to ship decisions can be taken more objectively
Conclusion

• Focused Iterative Testing is a general technique

• Very successful in testing multithreaded, multiprocess, multiuser enterprise server software.

• Beyond databases it could be applied to Messaging Systems like WebSphere MQ, journaled file systems like JFS in AIX, Enterprise Service Bus like WebSphere Message Broker etc.
Questions?