z/TPF EE V1.1
z/TPFDF V1.1
TPF Toolkit for WebSphere® Studio V3
TPF Operations Server V1.2

IBM Software Group

TPF Users Group Fall 2005

z/TPF Test, Drivers, Automation and Performance

QUALITY Is Our Goal

Name: Kevin Jones

Venue: Main IBM Presentations

AIM Enterprise Platform Software

IBM z/Transaction Processing Facility Enterprise Edition 1.1.0

Any references to future plans are for planning purposes only. IBM reserves the right to change those plans at its discretion. Any reliance on such a disclosure is solely at your own risk. IBM makes no commitment to provide additional information in the future.

Agenda

- Introduction
- TPF and z/TPF Development Process
- Measuring Quality
- TPF and z/TPF Test Team
- Test Tools
  - Drivers
  - Automation
- Release-level Testing
- Conclusion and Invitation

Introduction

- IBM develops and implements emerging Software Test disciplines
  - Technical skills
  - Best practices
  - Career paths
  - Tools
- IBM invests significantly in the on-going test efforts for TPF and z/TPF
  - People
  - Tools
  - Scope

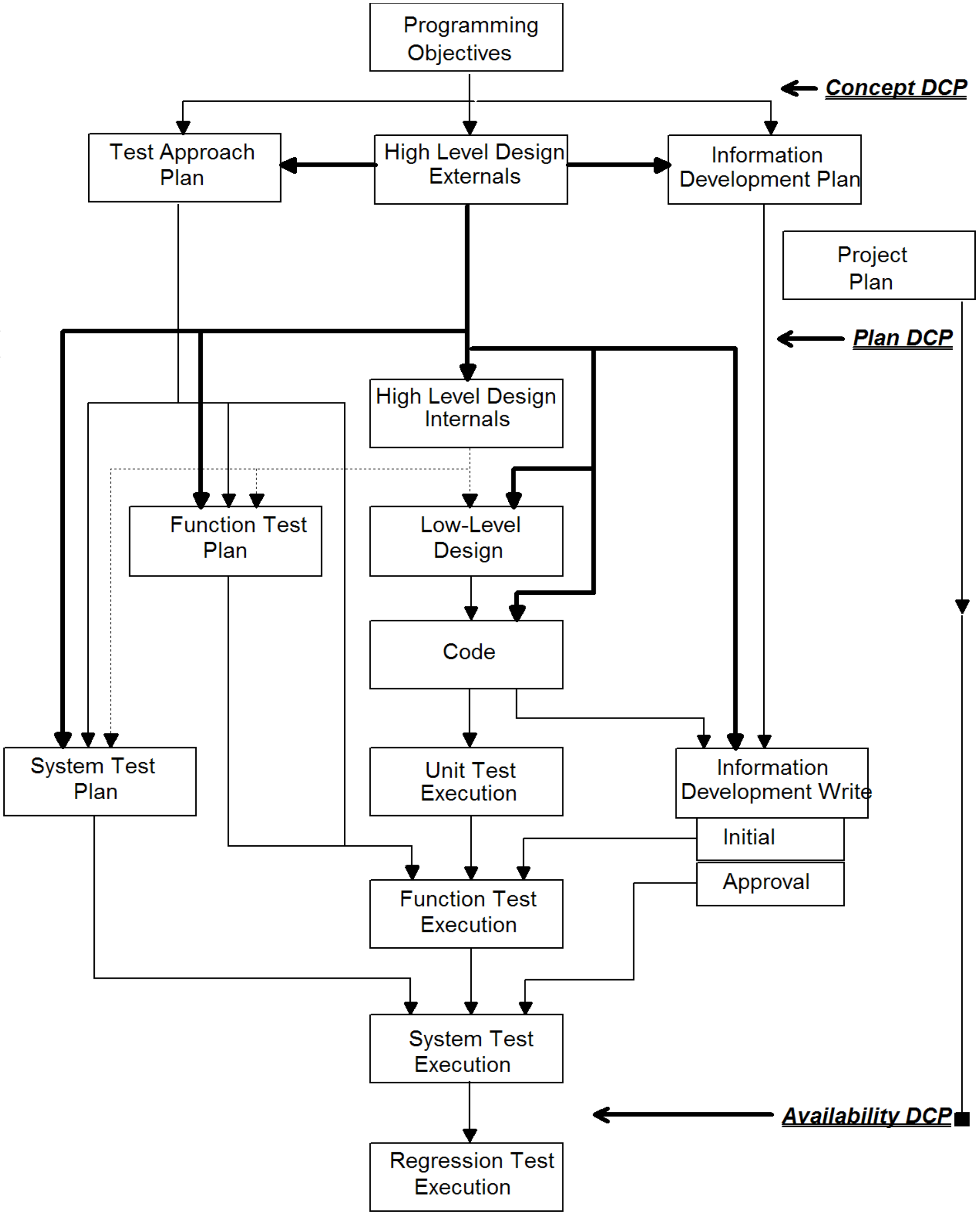
Introduction *(continued)*

- Software Test is not an after-thought to satisfy a development process. It an important discipline used to *improve quality*, which then:
  - improves customer satisfaction
  - reduces maintenance requirements
  - allows a focus on the development of new function

Measuring quality and resolving areas of concern is an important part of the TPF and z/TPF development process. The same disciplines can be used in any software development process

TPF and z/TPF Development Process

- 15 process stages, including:
  - four design stages
  - *one* code writing stage
  - *four test stages*



TPF and z/TPF Development Process

- Unit Test
  - complete code coverage by forcing paths as needed, including error paths
- Function Test
  - Tests all new procedures and externals (for example, APIs and commands)
  - Normal operation and error conditions
  - Requires reviewed and approved test plans
  - *Key quality indicator*
    - *Quality is analyzed throughout function test*
    - *Mitigation plans are implemented as needed*

TPF and z/TPF Development Process

- System Test
  - Verifies projects when introduced to:
    - System workloads
    - *24 x 7 x 365* environment
    - Other projects on the same PUT
  - Requires reviewed and approved test plans
- Regression Test
  - Verifies all areas of the system on a regular schedule
  - Emphasis is placed on areas affected by recent maintenance (APARs)

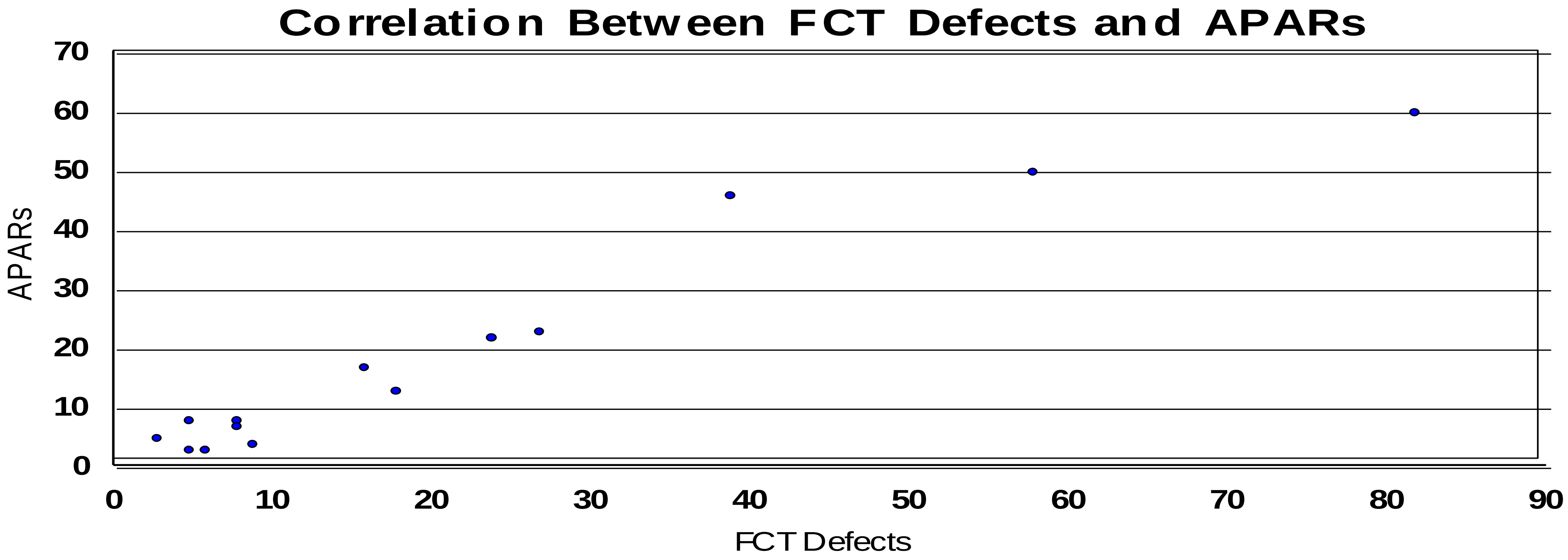
Measuring Quality

- *The TPF and z/TPF Development Lab measures quality using defect rates during Test, and takes action when flags are raised*

- *Measuring defects and taking action when needed is critical to the success of any software development project*

Measuring Quality *(continued)*

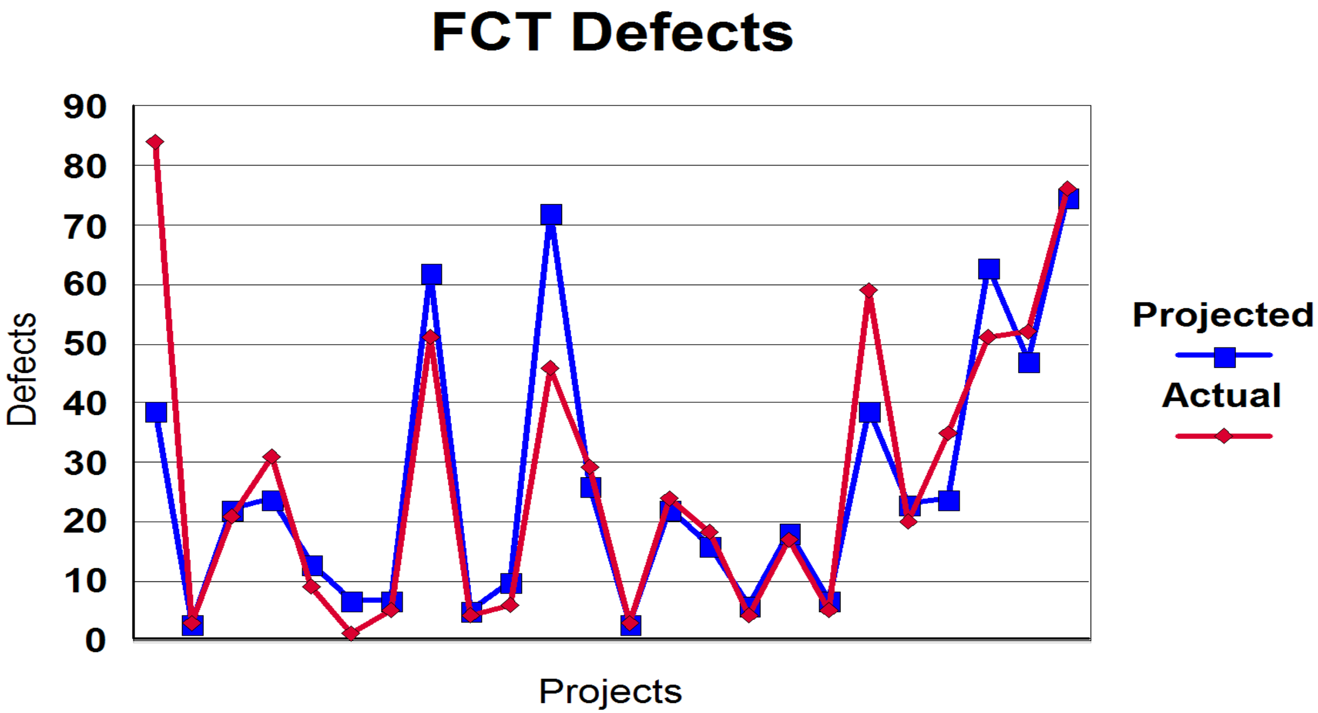
- Function Test defects are especially noteworthy
  - Data collected since the early 1990s shows that when a consistent process is followed, there is a strong correlation between Function Test defects and APARs



Measuring Quality *(continued)*

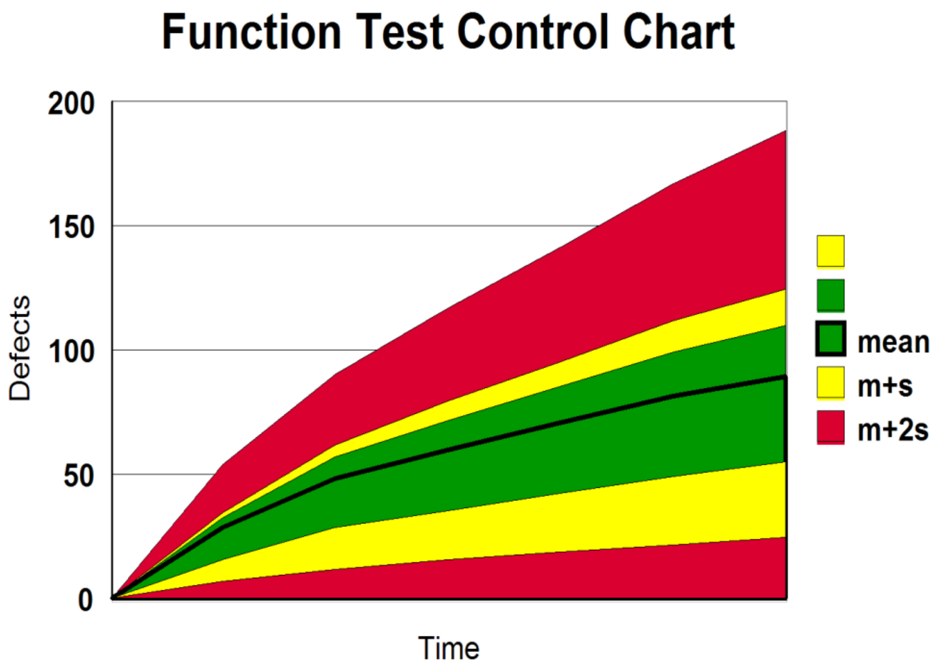
- Defects are projected after all externals (“High Level Design Externals”) are designed, using:
  - Project size measured by new and changed lines of code
  - Project complexity assigned by the Development team, and reviewed by Management
  - Defects must be valid, unique and APARable to be tracked by this analysis
  - Projections are determined by Bob Blackburn, a PHD statistician working for the TPF and z/TPF lab

Measuring Quality *(continued)*



Accurate predictions ensure that projects with potential quality issues are correctly identified

Measuring Quality



- Defects are reviewed regularly
  - High rates may predict a high number of APARs
    - Mitigation may include additional testing, or returning to an earlier development phase
  - Low rates may indicate inadequate test plans
    - Mitigation may include addition of test cases, or re-execution of the test plan by another team member

TPF and z/TPF Test Team

- The TPF and z/TPF lab includes a dedicated and experienced Test team responsible for:
  - – Development of test tools
    - Drivers
    - Automation
  - – Function, System and Regression Tests of all projects
  - – Release-level testing for TPF and z/TPF
  - – Identifying defects and tracking resolution
  - – Measuring quality to ensure continuous improvement

TPF and z/TPF Test Team *(continued)*

– Verifying TPF and z/TPF compatibility with new hardware and software

– Maintaining test systems for use by all teams

- *“24 x 7”* systems
- *Sandbox* systems

Early involvement by the Test Team in the Development process is critical

Test Tools - Drivers

- A *driver* is a collection of z/TPF realtime programs written to test specific z/TPF functionality
- Our driver suite has evolved over decades and now includes over 60 drivers testing more than 95% of all APIs
- The driver suite increased by over 40% while testing z/TPF, and many existing drivers were updated
- Driver development uses the IBM TPF Toolkit for WebSphere Studio

Test Tools – Drivers *(continued)*

- Drivers are critical to the success of our test efforts since they:
  - provide an interface to test APIs (macros and functions)
    - This interface can then be accessed by automation allowing tests to run automatically
  - test entire areas of the system (for example, CPU scheduler) with a few commands

Test Tools - Automation

- Automation complements drivers by performing repetitive tasks automatically:
  - Test drivers
  - System utilities (for example, PDU)
  - System maintenance (for example, maintaining a pool of scratch tapes)
  - System monitoring (for example, sending e-mail alerts when systems become unavailable or degraded)

Test Tools – Automation *(continued)*

- Our automation suite has rapidly expanded over the last 2 years to address many areas of z/TPF test, system maintenance and system monitoring
- Automation is based on the TPF Operations Server (TOS)
  - Scripts are written as:
    - *VAR* files written in REXX, ***or***
    - *TDR* files which include simple scripting options
    - *FIL* files which include simple lists of z/TPF commands

Test Tools – Automation *(continued)*

- Automation is also critical to the success of our test efforts by providing:
  - a consistent and repeatable interface to z/TPF commands
  - consistent and wide-ranging test coverage
  - a platform for maintaining systems and reducing human error
  - a platform to run *24 x 7 x 365* test systems

Test Tools – Automation *(continued)*

- – a method of running tests repetitively
  - human intervention required only if errors occur
    - – even knowledge on how to respond to common errors can be automated
- – a means for the TPF and z/TPF Development Lab to focus on new areas where defects are more likely

Automation is much more than just a means to reduce operation costs. It is used to drive all aspects of testing z/TPF including running test plans and drivers, and maintaining and monitoring systems.

- *Several z/TPF drivers are available for download, with plans to add many automation scripts and more drivers!*
  - http://www.ibm.com/tpf/download/tools.htm
  - Drivers and automation scripts are provided “as is” without any warranty or support
  - Many drivers and scripts will require modification or system definitions (for example, fixed file records) to run in your test environment

Learn more tonight at the IBM Hospitality Suite and tomorrow at the Systems Control Program subcommittee! Attend the Hospitality Suite and receive a free gift (while supplies last) – a USB memory stick containing sample drivers, automation scripts and other surprises!

Release Level Testing *(continued)*

- Release Level Tests are specialized tests designed to find defects that may not occur in Function and System Tests:
  - Migration, Coexistence and Fallback
  - *24 x 7* System Test
  - *Enhanced Error* testing
  - Field (“beta”) Tests

More details on these topics at the Systems Control Program Subcommittee tomorrow

Release Level Testing *(continued)*

- Performance Tests
  - Major emphasis for z/TPF
  - AIR1 driver accurately simulates customers workloads
  - Also focused on the performance of C/C++ by testing middleware such as Websphere® MQSeries, Mail and SOAP/XML
  - Significant system changes were implemented as a direct result of these tests

z/TPF Performance Statement

IBM estimates that z/TPF and z/TPFDF will, on average, result in a 10% increase (factor of 1.1) in CPU utilization as compared to TPF and TPFDF. This applies when comparing equivalent applications, message rates and use of operating system functions. Performance is influenced by many different factors, so every z/TPF and z/TPFDF system will be impacted differently. Your results may differ from this estimate depending on your system's profile.

Conclusion

- IBM and the z/TPF lab are dedicated to improving software quality through test disciplines
- Measuring quality is key to a successful software development project
- Test drivers and automation have significantly improved our ability to identify defects

Join Us to Learn More!

- IBM Hospitality Suite
  - Tonight, from 7:00 pm to 10:00 pm
  - Live demonstration of our Drivers and Automation
  - Ric Hung, Judy Kearney and Kishore Nagareddy will be available from our Driver and Automation team

- System Control Program (SCP) Subcommittee
  - Tomorrow, from 11:30 am to 1:00 pm
  - A more detailed, technical presentation

Question and Answer