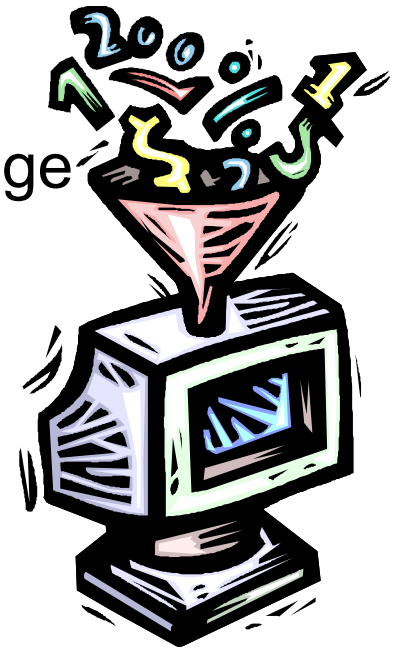
SAND2007-3452C

Evolution of a Test Capability

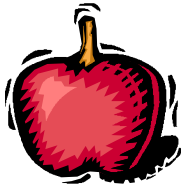
for an ARS-Based Trouble-Ticketing System

An Overview of the System

- Remedy Action Request System (ARS) 6.0.3 and Enterprise Service Suite (ESS) for ARS 6.0.3.
- ~1000 individual users & entity accounts
- ~130 teams (2-3 teams added/mo. since go-live)
- Users per Service Offering: Heavy: ~290, Medium: ~135, Lite: ~475
- No. Tickets created per day: ~720 on an average day, ~1700 on a heavy day.
- No. Tickets closed per day: ~760
- No. Tickets open at any given time: ~4100

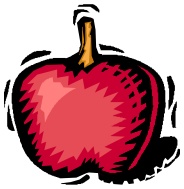


Differences Between Old ARS System & ESS



- Configurable in one dimension vs. configurable in three. (Can you say, “more testing scenarios”?)
  - Service Offering (Lite, Medium, Heavy)
  - Lite and Heavy Consoles
  - Team-Specific Configuration Options (12 of them!)
- Easier to add new teams→ more users→ more people affected by problems we don’t find during development.
- Heavy & Lite consoles vs. separate forms or applications.

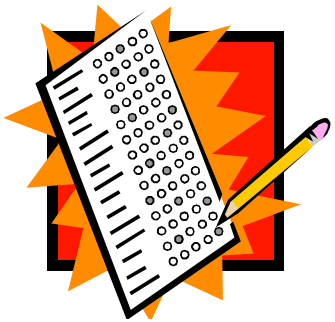
Differences Between Old ARS System & ESS, cont'd



- More complex functionality. (Can you say, "more testing scenarios"?)
  - Ability to have only one child ticket open at the time vs. ability to have more than one child ticket open at the time.
  - Ability to have child tickets and appointments in series vs. ability to have child tickets and appointments in parallel.

The Old Approach to Testing

- Basic scenarios were documented, but not all ways were tracked/checked if there were multiple ways to get there.
- Systematic testing was done primarily prior to new implementations.
- Regression testing did not exist.
- “State of the System” = ???



Challenges

- Loss/change of development personnel.
- The system is highly configurable, but the trade-off is that every place a function can run must be tested individually.
- Time!

Changes

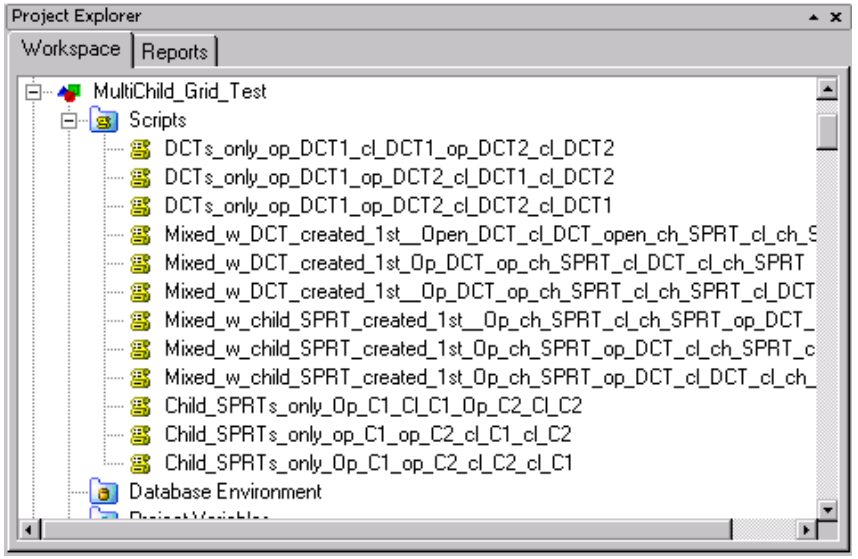
- Bring on a person to do testing full-time (i.e. yours truly)
- Pick up an automated testing tool to speed up the testing.

The Approach to Testing

- Methodical, logical, and more than a little paranoid.
- Take anything the developer tells you about how the system is working with a grain of salt.

The Testing Grids

- The Work Being Done grid
- The Cross-Service Offering grid
- The Reassign/Own It grid
- The Notification grid
- The Bug grid (planned)



The Testing Product & Why Chosen

- QA Wizard from Seapine Software
- Good support—they were the only ones who offered support during the trial period.
- A product for testing by doing; (you don't have to be a programmer to use it!)
- Able to test both Windows and Java apps.

How It's Helped So Far

- We can find problems more proactively.
- Faster grid-testing.

The Future

- To develop a complete set of tests and run them periodically in order to determine the “state of the system”.
- To have trackable releases for enhancement requests.

QUESTIONS

Thanks for your time!