

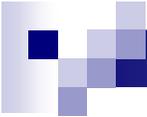
# Evolution of a Test Capability

**for an ARS-Based Trouble-  
Ticketing System**

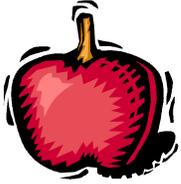
# An Overview of the System

- 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 on an average day, ~880 on a heavy day.
- No. Tickets open at any given time: ~4100





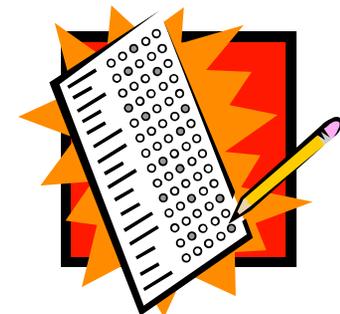
# Differences Between Old ARS System & ESS



- 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!)
- More users who are less technically sophisticated.
- More complex functionality. (Can you say, “more testing scenarios”?)

# The Prior Approach to Testing

- Basic scenarios were documented, but not all ways were tracked/checked if there were multiple ways to get there.
- Thorough testing was done prior to major implementations but only spot-testing was done for smaller changes.
- Thorough testing included role-playing, “get acquainted” sessions, and unstructured testing.
- 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!



# The Approach to Testing

- Methodical & logical
- Take anything the developer tells you about how the system is working with a grain of salt.



# Changes

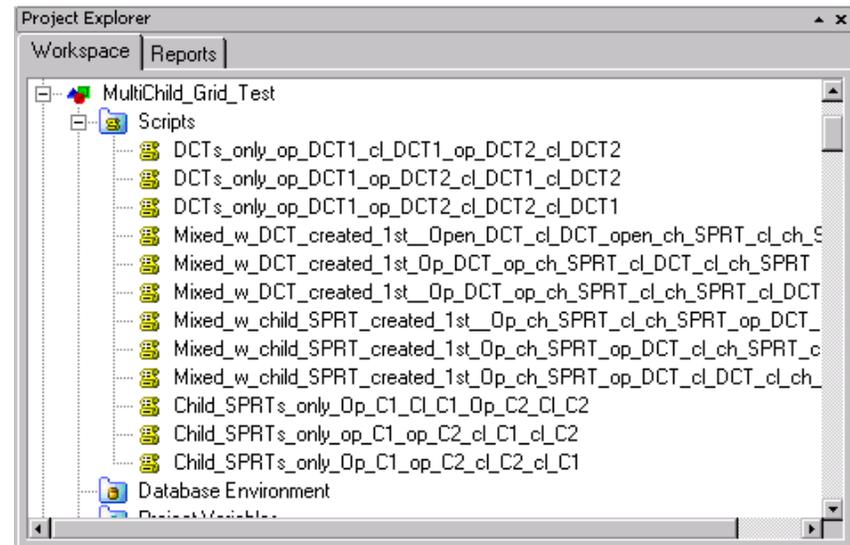
- Bring on a person to do testing full-time
- Pick up an automated testing tool to speed up the testing.

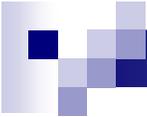




# The Testing Grids

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





## How It's Helped So Far

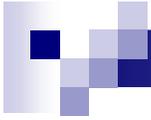
- We can find problems more proactively.
- Faster grid-testing.
- The product forces identification of the smallest unit of functionality for testing.
- Forces consistency of testing—the scenario is always tested the same way



# The Future

- To develop a complete set of automated 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!

