

Introducing the Mainframe Batch Maturity Model – What's Your Maturity?

Session ID: 25573
Pittsburgh, August 2019

Kelly Vogt
Compuware

kelly.vogt@compuware.com

Your Presenter



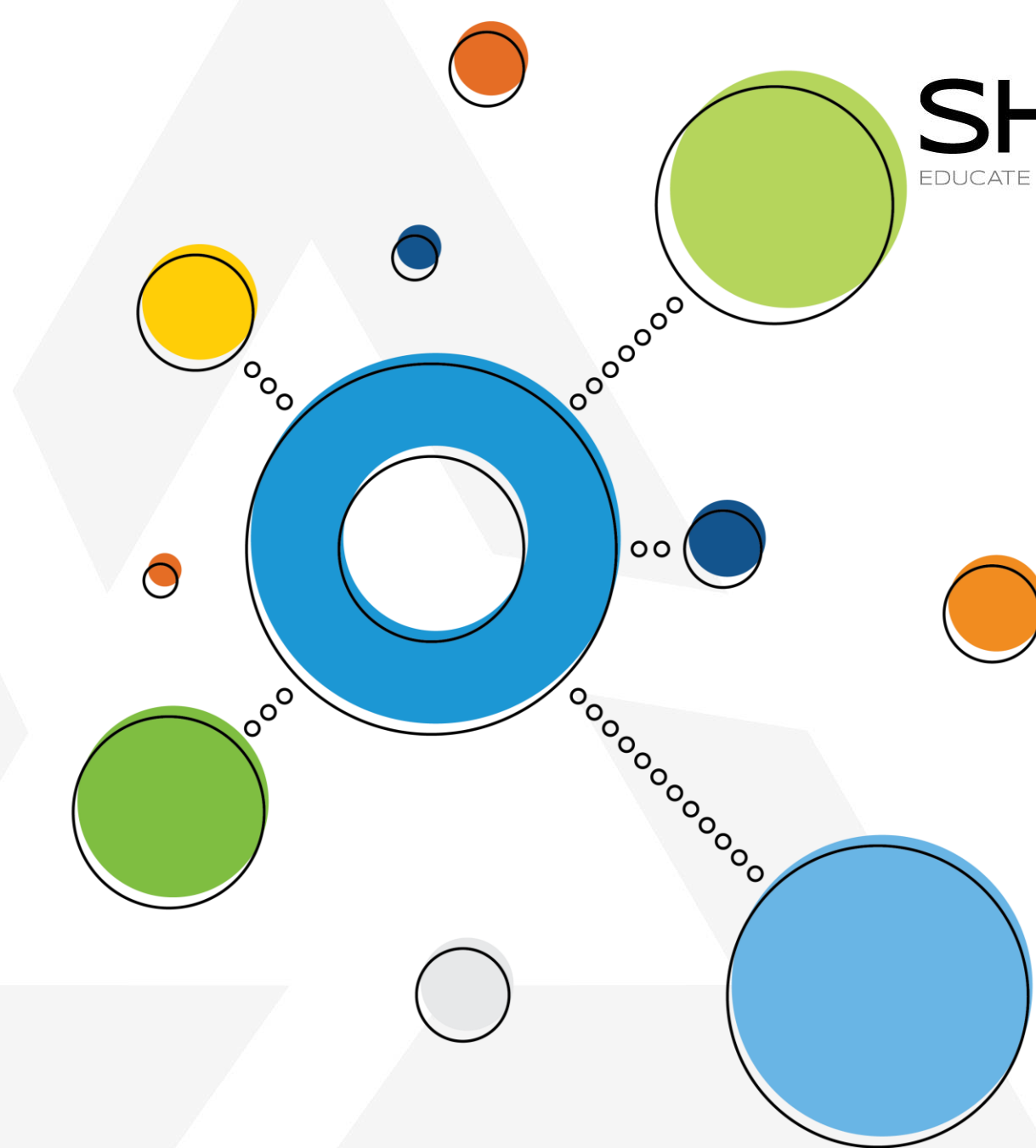
Joined Compuware in February 2018 as a Field Technical Support representative.

Previously, 38 years in mainframe systems programming, performance and capacity management and data center management.



Agenda

- Current State of Batch
- Batch Maturity Model
- Future State

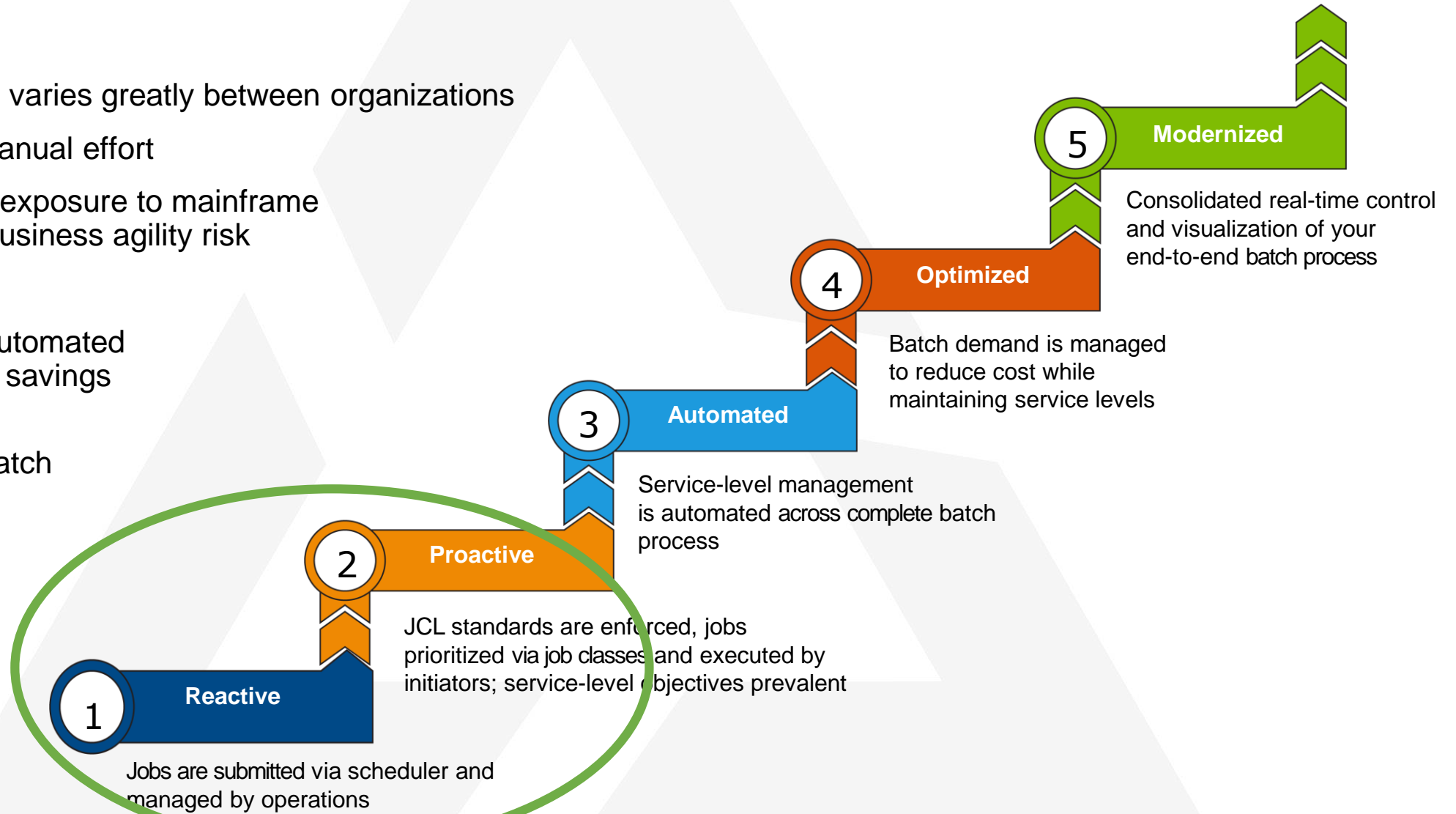


Current State of Batch

- **It's still a manually operated environment!**
- Operation of JES2 is not for the faint-hearted
 - Much knowledge required
 - Juggling and plate-spinning skills required
 - Domain knowledge required
 - Constant eye-on-the-ball lest you eat a line drive
- Few shops can just set up a structure... and it works all the time
 - And if you are, are you measuring how well its working?
- Problem resolution requires experience to keep batch moving in a timely fashion

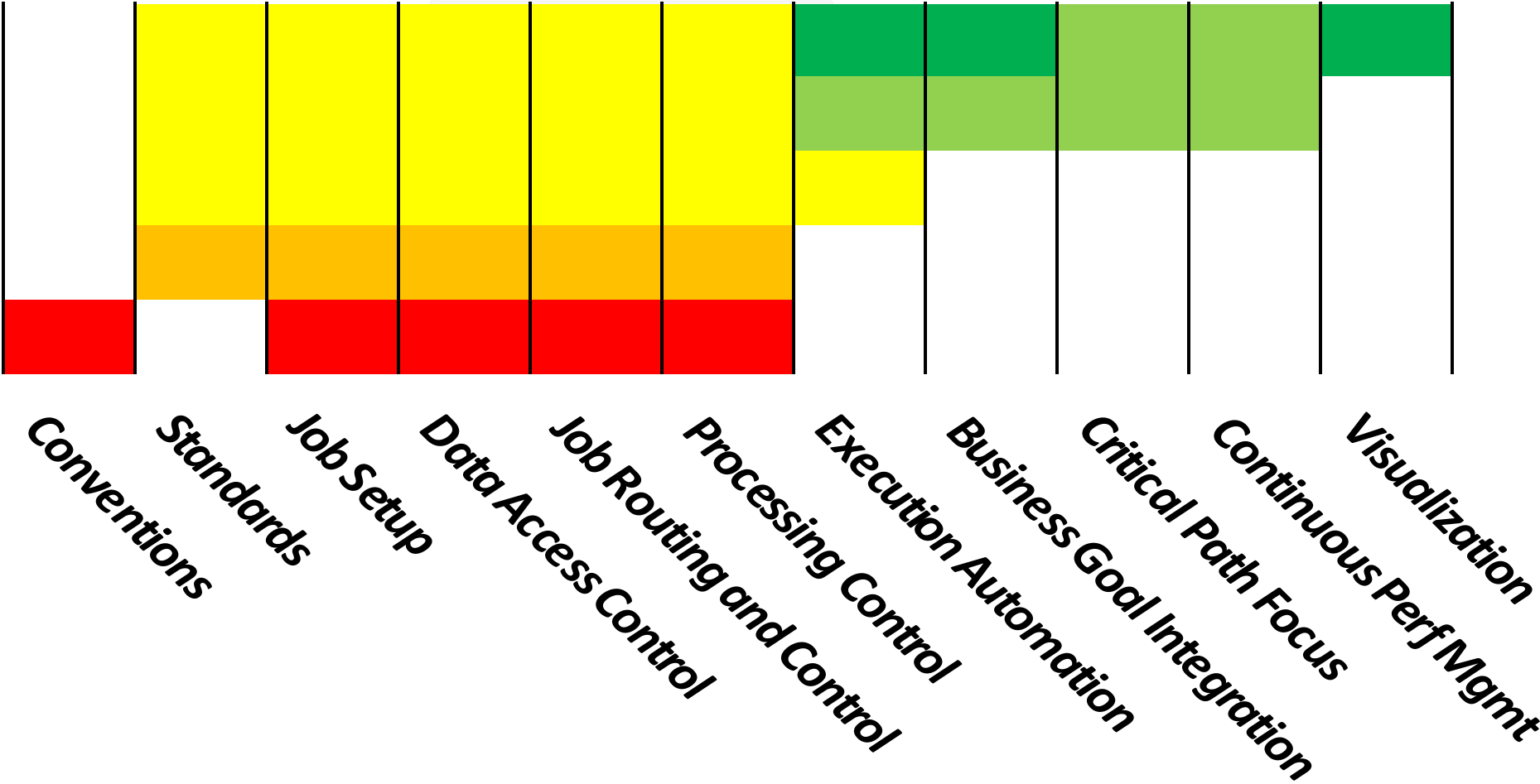
Mainframe Batch Maturity Model

- Level of maturity varies greatly between organizations
- Lower = more manual effort
- Lower = greater exposure to mainframe retirement and business agility risk
- Higher = more automated and greater cost savings
- Higher = better future-proofed batch



Maturity Model Components

Visualized
Optimized
Automated
Proactive
Reactive



Level 1: Reactive

- Check your pulse – you've been through here...
- This is vanilla JES2



L1: Conventions

- Budding Standards
 - Job names, PROC names, job step names... ?
 - Program names, DD names, dataset names... ?
 - JOB, EXEC, DD, OUTPUT parms... ?
 - Tremendous flexibility... to a fault
- Published – at some point... ?
- Enforced by the honor system
 - Problematic at least. Potentially disastrous!
- New technology
 - May not be exploited
 - May cause problems



L1: Job Setup

- Migrated datasets recalled during job execution
 - Elongates JOB wall-clock time
- Migrated Virtual Tape Volumes (VTVs) staged during job execution
 - Recalled when mounted
 - Elongates JOB wall-clock time
- Physical tapes?

L1: Data Access Control

- Dataset contention occurs frequently
 - SYSDSN
- Volume contention arises from stacked datasets on tape
 - SYSZVOLS
- **Schedule disruptions** may cause additional contention situations
- Operator on the hot seat
 - Which job(s) do I CANCEL?

L1: Job Routing and Control

- Jobs routed *where* required resources exist via SYSAFF
 - What happens if the resource is not available?
 - Manual handling?
- Operator or job scheduler controls *when* they arrive
- What if you need to move a data base region?
 - JCL changes required...
 - Scheduler changes required...
 - How to communicate, coordinate, test?
- How to manage database maintenance? Rolling IPLs?

L1: Processing Control

- Concurrent executions by users uncontrolled
- It's the Wild West
- Interdepartmental dramas break out
 - Operator/management must arbitrate
- Over-initiation tendency
 - Job wall-clock time elongates
- Under-initiation
 - Wasted opportunity to run more work



Level 2: Proactive

- Business needs force evolutionary progress
- Deadlines **MUST** be met!
- Objectives are established
- Cleverness is the order of the day

L2: Standards

- Enforcement of conventions yields *STANDARDS!*
 - JES2 job-related exits
 - TSO submit exit
 - SMF exits
- The penalty for all infractions is death
 - Job cancellation *so/ves* problem jobs
 - “Our Way or the Highway!”
 - Whack a mole
- A Cold War between Infrastructure and AppDev
- Advancement slowed if not stalled

L2: Job Setup

- Users *proact* by ensuring their datasets are always at hand
 - Methodically *touch* data to keep it from being migrated
 - Wastes CPU cycles
- Jobs scheduled to force HSM recalls
- IEFBR14 job steps – Critical, but...
 - Recall migrated datasets from HSM to delete
 - Stage VTVs to cache to delete
 - Waste of clock time, CPU time and I/O

L2: Data Access Control

- Late batch may precipitate dataset contention
- Scheduler features might be used to avoid events
 - Negative job dependencies
 - Must be documented
 - Must be maintained when conditions change
 - May be ineffective over time
 - May become a hindrance over time

L2: Job Routing and Control

- Scheduling environments may be exploited
 - WLM changes required
 - Resource name states must be set timely and accurately
- Multiple resource requirements complicate usage
 - Increasing number of permutations required
- Users must understand, keep informed and specify correctly

L2: Processing Control

- Initiator CLASS schemes grow more complex – often unwieldy
 - Managing parallel job executions adds to job class creep
 - *Departmental* job classes
- Users must understand, keep informed and specify correctly
- Misuses must be detected and controlled
- Operators must monitor and manage
 - Proper initiation levels difficult to control

Level 3: Automated

- Policy-based controls emerge
- Service Level Agreements provide initiation prioritization
 - Time to initiate!
 - Job business importance!!
- Human intervention only for exception handling
- Policy yields agility - enables immediate solutions to problems

L3: Standards

- Standards enforcement become dynamic
- No longer dependent on JES2 internals and Assembler knowledge
 - ISV solutions normalize complexity
- Gaming the system – ended
- Jobs corrected when possible
 - Job cancellation is a last resort...

L3: Job Setup

- Migrated datasets/archived volumes are recalled/staged automatically and optimally
 - Prior to job initiation
 - Prioritized by job importance/urgency to initiate
- Wasteful, unnecessary recall/staging requests are avoided
 - Why recall a dataset or mount a volume only to delete it?
- Job setup delays are measured and recorded!

L3: Data Access Control

- Access to data is managed by business importance/urgency to initiate
 - Not 1st come – 1st served
- SYSDSN and SYSZVOLS contention is automatically avoided
 - Job dataset and volume requirements, unit requirements known
 - Inter-job relationships – contention not allowed to occur
- Remaining contention handled by automation
 - Operator intervention minimized
- Data availability delays are measured and recorded

L3: Job Routing and Control

- Resource requirements automatically detected
- Jobs automatically routed to where and when resources are available
 - Jobs held until resources are available
- Conflicts are resolved automatically without operator intervention
- Routing delays are *measured and recorded*

L3: Processing Control

- Processing control is abstracted from job class
- Ability to arbitrarily group and sub-group jobs
 - Access to resources controlled independently and automatically
- JESplex and/or LPAR-level
- Processing control delays **measured** and **recorded**

L3: Execution Automation

- Automated initiator control
 - INITs started and stopped as needed
 - System performance dictates initiation levels – neither over- nor under-initiated
 - Static, pre-planned INIT structures give way to dynamic, governed INIT management
- SLAs control job selection
 - Job queues reordered by SLA to initiate and job business importance
- Production is preferred over non-production
- Operator is hands-off
- Performance against SLAs is **measured** and **reported**

Level 4: Optimized

- Minimized operational costs
- SLAs not sacrificed
- Licensing model awareness
- Program-level awareness
- Critical path is primary production batch driver

L4: Execution Automation

- Cost containment goals managed automatically
 - Demand management for CPU resource
 - Demand management for expensive/limited software
- Batch completes on time
 - Neither after nor before schedule
- CPU consumption balanced across LPARs

L4: Business Goal Integration

- Business operational costs rationalized
- Business becomes informed of its impact on processing costs
- Jobs become 'business decisions'
 - Is it worth paying more to run this job at this time?

L4: Critical Path Focus

- Application batch is understood from the job up
 - Job dependencies and interrelationships are known
 - Applications/job flows/streams interrelationships are known
 - Data used to project critical path
- Relative importance between entities is known
 - Drives job selection urgency
- Critical paths are known and prioritized
 - Execution managed to complete on time
 - Adherence is monitored and tracked
 - Deviations are detected; adjustments are made to stay on schedule, if possible
- Rerun allowances don't just happen – are computed and managed

L4: Continuous Performance Management

- Urgent/critical path job performance is compared to history
 - Significant clock-time/CPU time increases/decreases are automatically investigated
 - Measurement data is automatically captured
 - Drives performance tuning of programs
- Program performance/cost management driver!
 - Poor program performance is a defect
 - Why not high cost to operate?

Level 5: Modernized

- The future has arrived
- Reactivity is minimized
- Exception driven
 - Detect
 - Determine root cause
 - Create automatic solution
- SLAs consistently met
- Customers are informed, engaged
 - Batch becomes transparent to the Business

L5: Execution Automation

- Information drives continuous improvement
 - Job delay data, history, trends
- History data informs dynamic critical path management
 - Projections
 - Seasonality
 - Day of month/week, time of day
- Known exceptions proactively handled
- Machine Learning / Artificial Intelligence potential !

L5: Business Goal Integration

- Performance against goals transparency
 - SLAs performance graphically reported
- Application, job-flow, job-level critical condition alerting
 - Feed ITSM Event Management processes
- Job completion data linked to business processes
 - Business Process Monitoring becomes possible

L5: Visualization

- The 'green screen' is gone
 - Modern presentation layer for most activity
 - Intuitive, customizable and mobile
- Batch information de-mystified
 - Instrumentation data made easily consumable by tech and business alike
- Scheduler, system monitors, TSO and SDSF-like interfaces follow

Future State

- Batch MUST evolve – it's unwieldy
- Evolution is essential to health and efficacy
- Even more pressure to reduce batch windows
 - Little investment to rewrite apps to run against batch
- Retirements mitigation is critical
 - Risk reduction!
- Automation is *the* answer

Future State

- Embrace the model
- Most of this is achievable NOW with commercially available products
- Unplanned loss of experienced people can cripple your operation!

Future State

‘The more mature your batch processing capability, the less you will have to worry; the fewer fire drills you will suffer; the fewer carpet calls there will be where you play the starring role.’

Thank you!

Complete your session evaluations for a chance at daily prizes!

To complete, visit
www.share.org/evaluation
and see your progress on the
leaderboard!

