



Improving Validation Results by Automating Test Plan Creation

Presented by Kent Lohrey
Validation Program Manager
Total Systems Design



Automation of Test Content

- Business need
- Automation
- Demonstration
- Results/benefits
- Discussion/questions



Business need



Validation success depends on many factors

- Clear & measurable requirements
- Complete & accurate specifications
- Accurate test plans/protocols
- Experienced test personnel
- Prepared test environment
 - Installed system
 - Field devices
 - Utilities



Validation projects face common challenges

- Changes to requirements or design
- Inputs from many sources and/or companies
- Unpredictable testing environment
- Delays in validation predecessor tasks
- Deadlines that don't change



TSD manages challenges by focusing on success factors in our control

- Areas most often in TSD control
 - Design specifications (DS)
 - Test plans
- Impact
 - Any flaws in these documents result in extra, non-value added work
 - Minimizing issues in these areas maximize capacity to address issues outside of our control



Automation



TSD document templates were redesigned in 2003

- Specifications changed from sentences & paragraphs to tables
- Table-based designs
 - More complete and detailed
 - Greater consistency of content
- Templates were designed to be consistent with GAMP-4 principles



Revised DS was successful on several projects from multiple customers

- Several user interface (UI) objects were 90+% of UI content
 - Data required was dictated by object type – not object details
 - Action object types: buttons, entry boxes, pick lists, etc.
 - Define actions, conditions, access, and range
 - Display object types: process values, animated valves, etc.
 - Define conditions and corresponding display
- Process definition was consistent
 - Data required was the same regardless of process details
 - Define actions in steps and how to get to the next step(s)



Consistent detailed design created opportunity for automation

- Design specifications are collections of Consistently defined discrete pieces of information
- Design objects can be grouped by type
- Object types are consistent across projects, clients, industries
- Testing is also defined/structured by object type – not object details



A compressed schedule and resources constraints demanded it

- Test plans were written by typing or copying/pasting specification content
- A test plan had to be written quickly when most validation staff was in the field testing
- Methods had to change to deliver a quality test plan on time
 - Specification data was moved to Excel
 - Formulas were used to combine the discrete data
 - Immediate impact was obvious - specification objects could be processed much faster

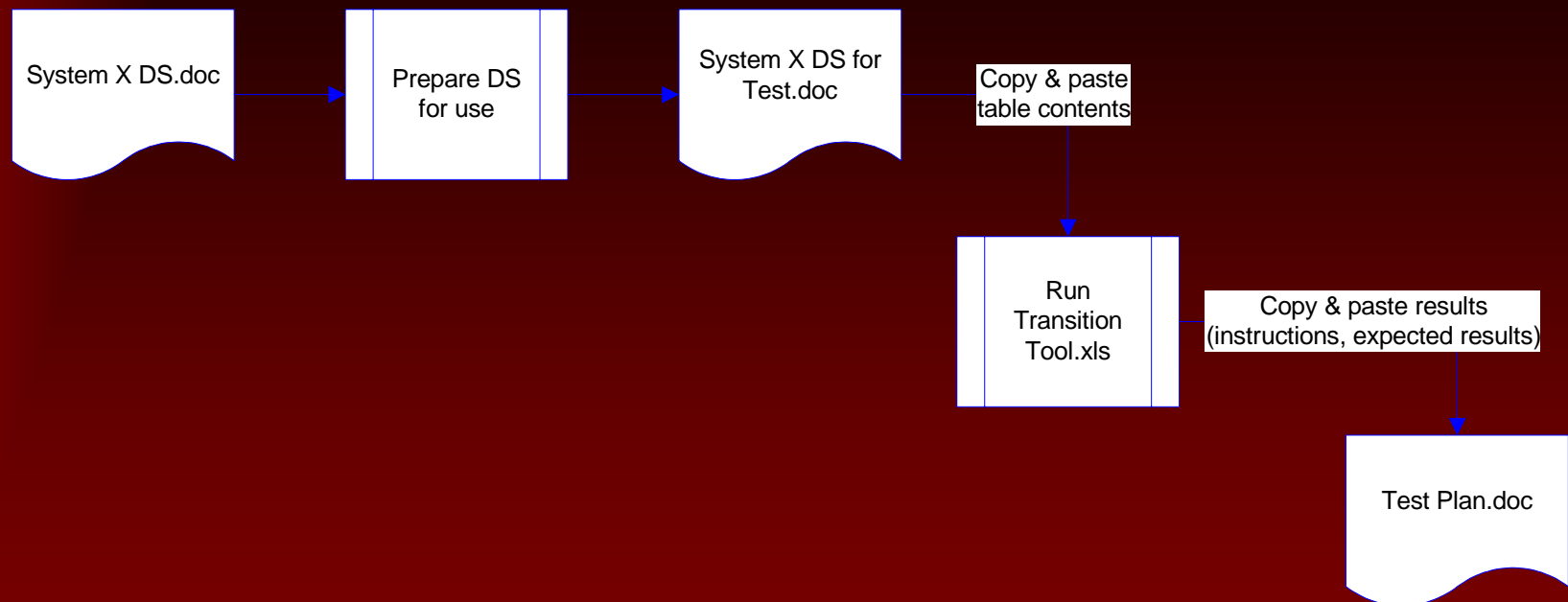


Positive impacts of this first transition tool demanded more

- Initial development effort was recovered before completing the first test written with the tool
- Loose collection of formulas has evolved into a mature method
 - A collection of macros applied within a defined process to move DS content into test documents
 - Defined process for handling data
 - Use of macros and functions for repeated tasks



DS content is processed in the tool, results are pasted into the test plan





Using this transition tool requires some structure

- Using table based DS
 - Tool is designed to match TSD DS template
 - Specific values in some fields trigger different logic in the tool
 - Has been used with table-based customer DS templates
- Following a structured process
 - Prepare DS document for use
 - Copy & paste DS data into matching tab in the tool
 - Apply formulas matching object types & run macros
 - Copy & paste results into test plan
 - Review for quality/clarity – edit as needed



Is this method or tool validated?

- The method and tool has not been validated
 - No written specifications or formal testing
 - Formal validation was considered unnecessary
 - Tool is only used to produce document content – not to control pharmaceutical/medical device production
 - Test content created is only draft – then edited & reviewed
- All changes are tested
 - Testing varies with the amount/type of changes
 - Major changes are tested by multiple engineers
 - Regression testing is performed



Demonstration



Automation is used to create user interface tests

(contact TSD for video demonstration)

Chocolate Chip Production Line

Main Menu

Security

100-XV-001 Water
100-XV-002 Flour

100-MX-01 Mixer

Level
100%
50%
0%

Roller Height

PV	### in
SP	### in

Roller Motor

Chip Type

Dark
 Milk
 White

Cookie Cutters

Chocolate Chip Cannon

Conveyor Speed

MON-DD-YY MON-DD-YY MON-DD-YY
HH:MM:SS HH:MM:SS HH:MM:SS

Production Mode

Automatic
 Manual

Alarms

Date	Time	Class	Type	Pri	Name	Group	Prov
15 May	10:48	Value	HHI	1	Alarm1	GroupName	Provider
15 May	10:48	Value	HI	250	Alarm2	GroupName	Provider
15 May	10:48	Value	LO	500	Alarm3	GroupName	Provider

100-TIT-01

Oven

Oven Temp

PV	## C
SP	## C
Output	## %

Conveyor Speed

PV	# FPM
SP	# FPM
Output	## %

Conveyor Motor

Dough

Start

Stop

Roller

Start

Stop

Cutters

Start

Stop

Chips

Start

Stop

Oven

Start

Stop

Conveyor

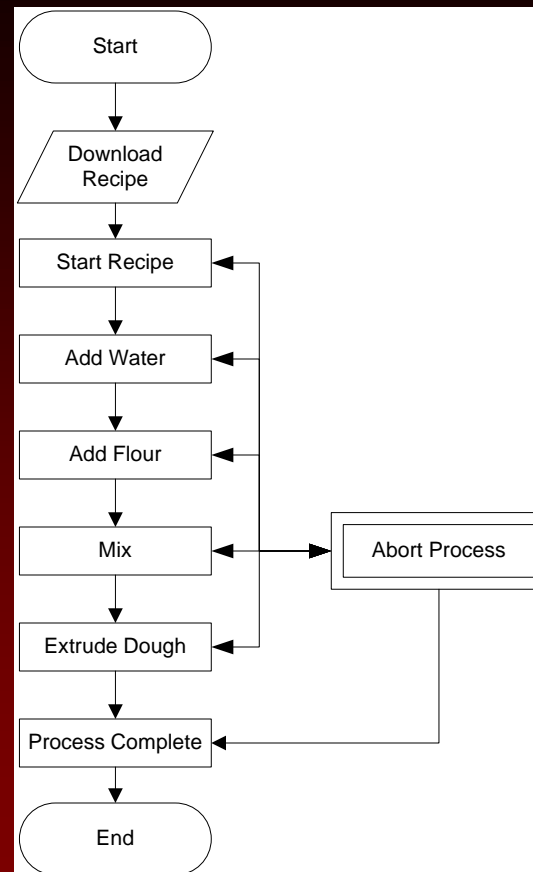
Start

Stop



Process test creation is also automated

(contact TSD for [video](#) demonstration)





Results/benefits



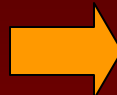
The Tool is a capability and productivity multiplier

- Automates many test writing steps



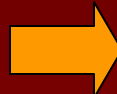
- Increases test writing efficiency/speed
- Reduces human error
- Allows more time for review
- Allows rapid response to DS changes

- Produces consistent test content



- Allows use of multiple resources
- Improves repeatability

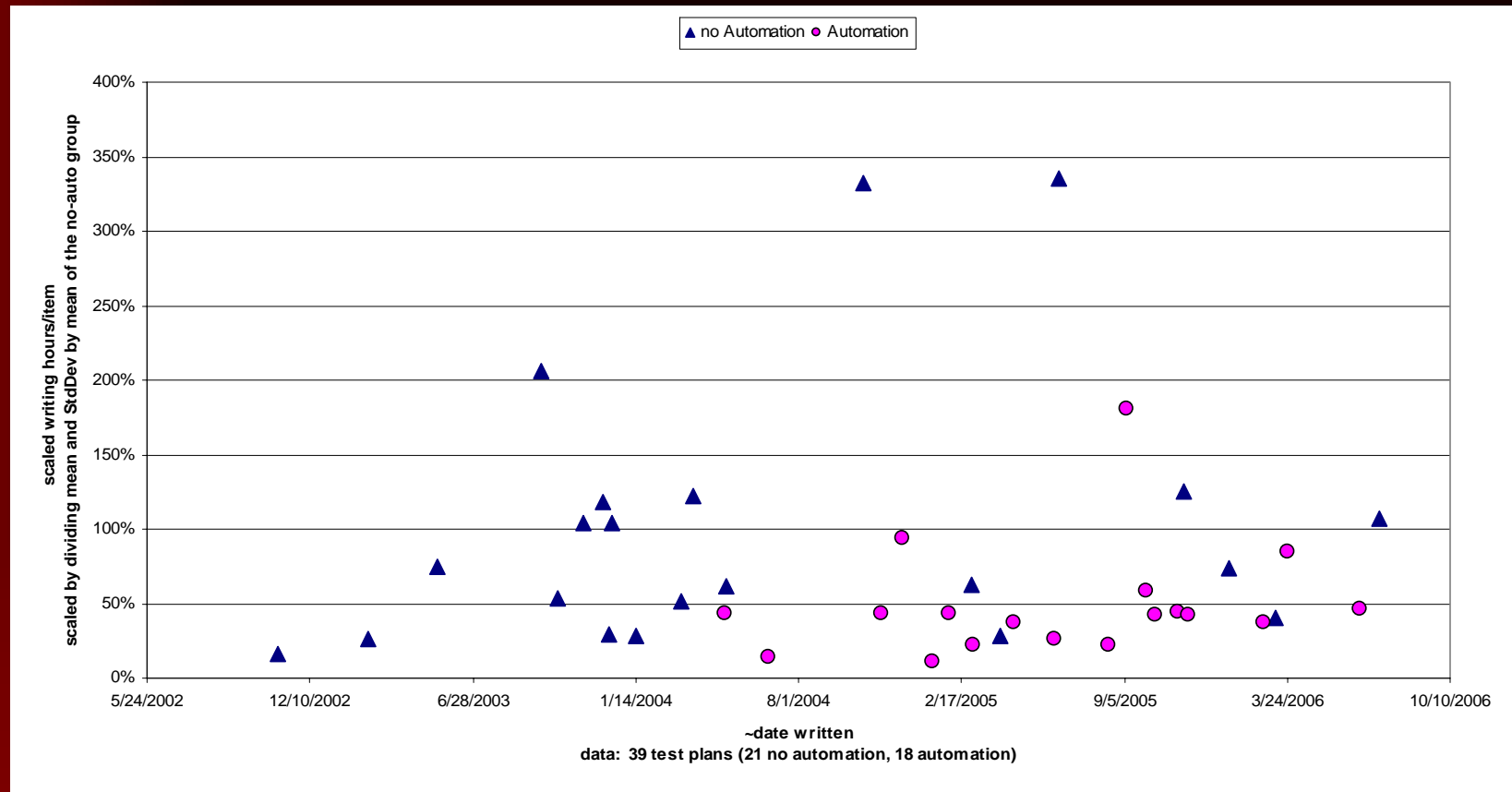
- Includes exact spec wording



- Reduces protocol generation errors/deviations

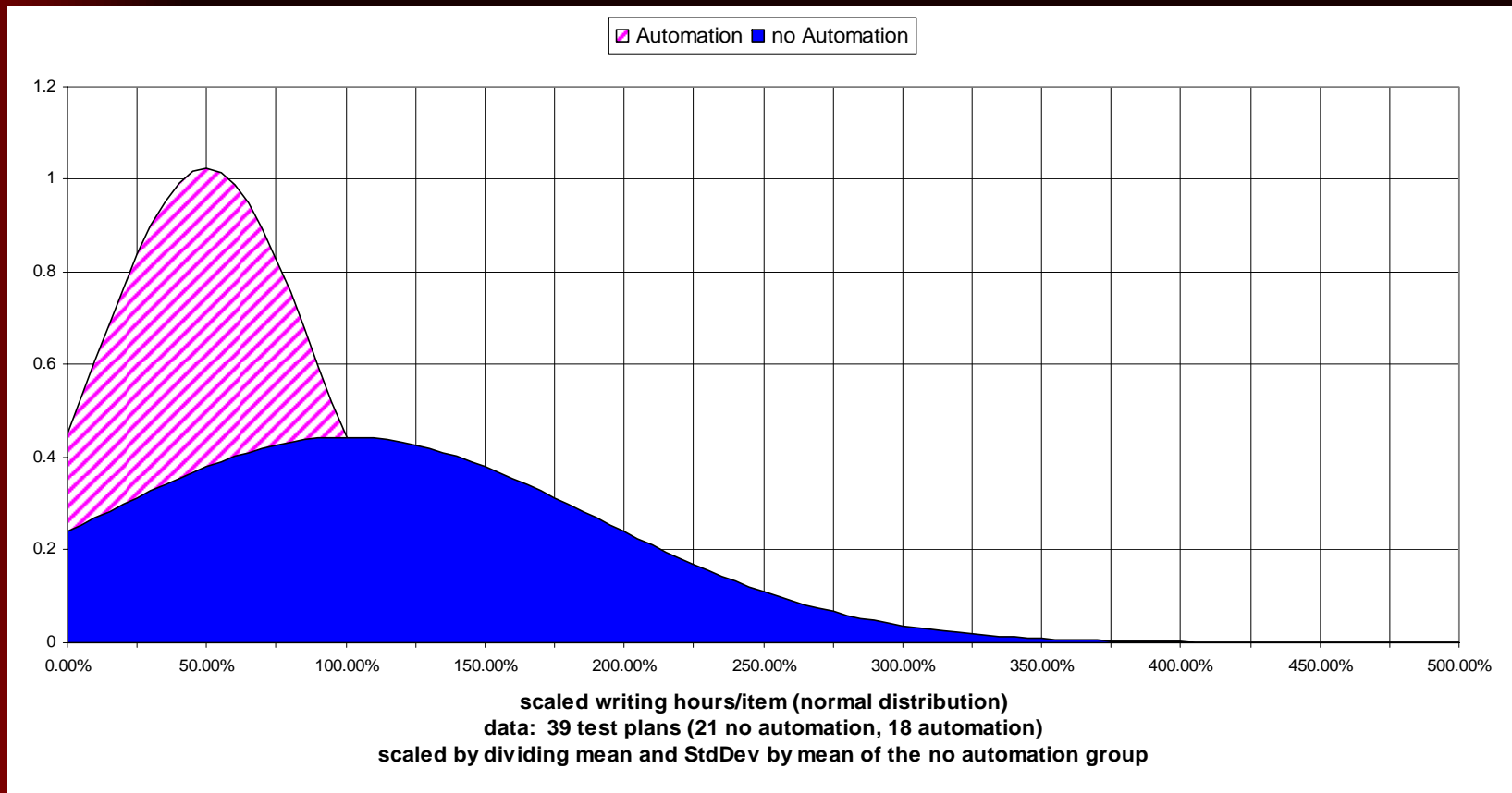


Test writing data shows ~50% reduction in writing time



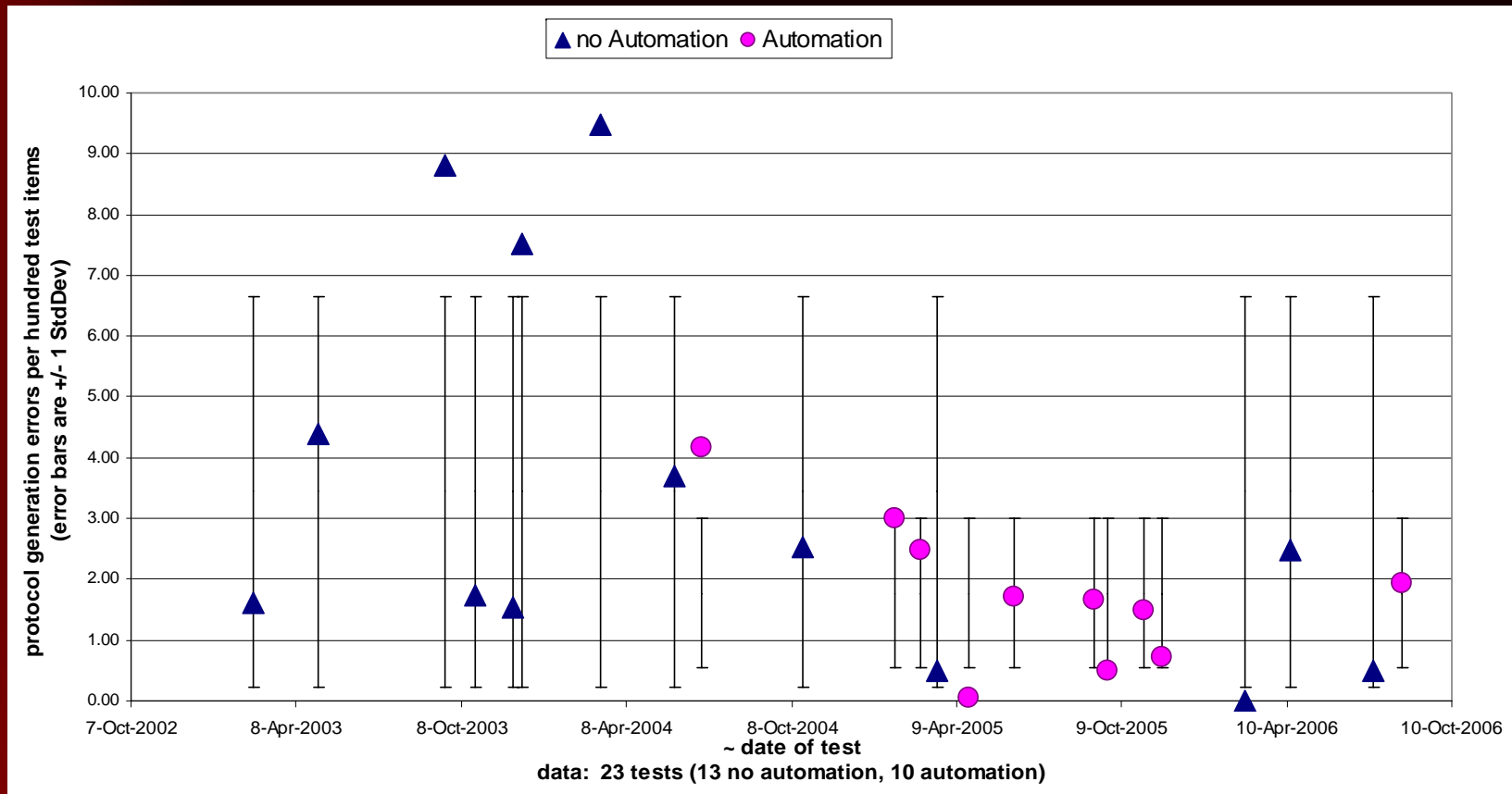


Faster test writing helps manage challenges like DS changes and schedule compression



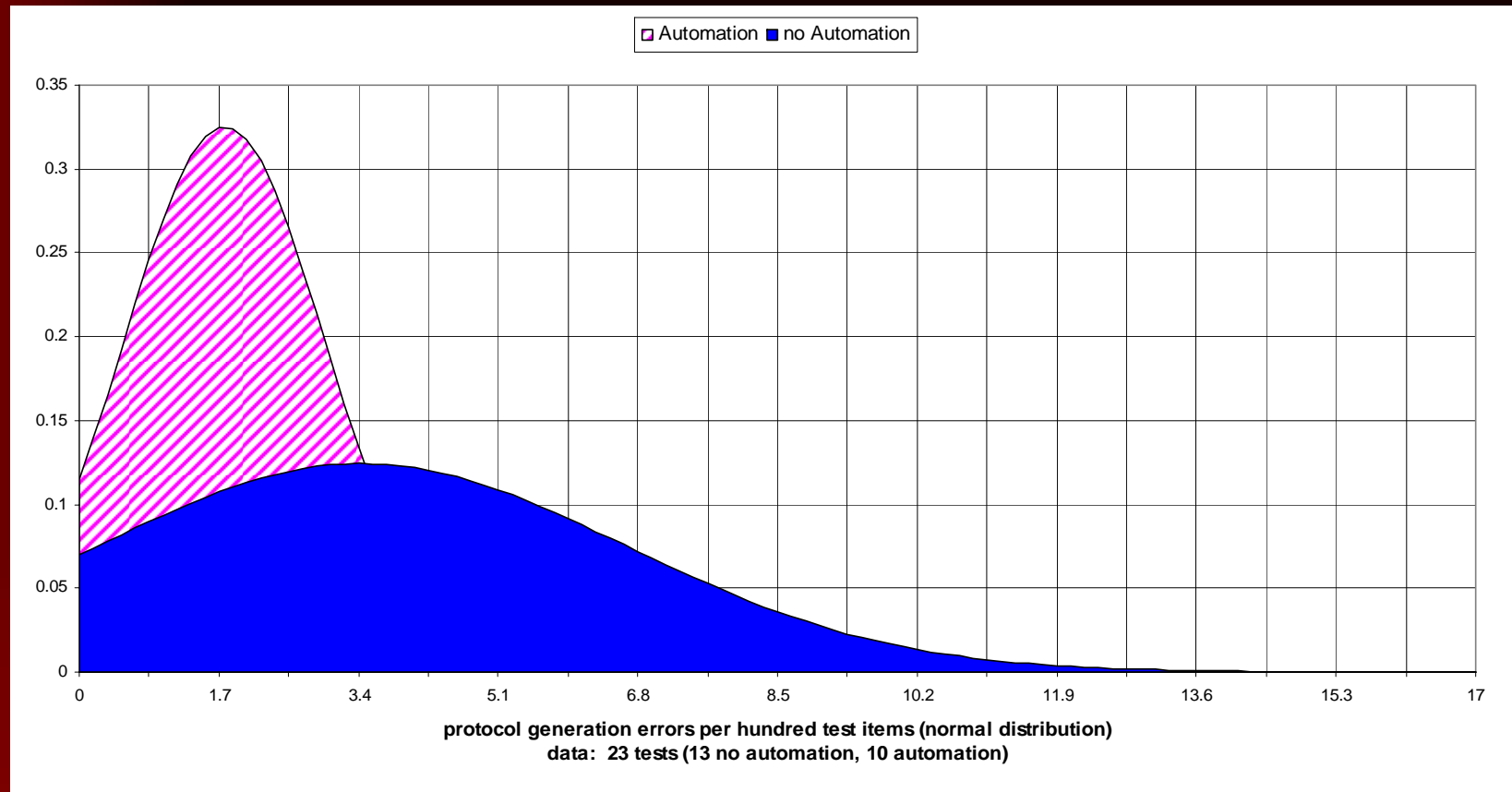


Test execution data shows ~50% reduction in protocol generation errors





Reduction in protocol errors reduces testing time and cost - helping to meet deadlines





Automation of test creation delivered additional benefits

- Demonstrated the benefits of productivity tools and process improvement to employees
- Increased understanding and experience with macros
- Generated ideas for other tools
 - Specification (DS) builder (alarms from I/O list)
 - Code builder (software tags from I/O list)
 - Specification (DS) to user manual



Discussion/questions