

**MODEL-BASED
QUALITY ASSURANCE OF
MICROSOFT PROTOCOL
DOCUMENTATION
ILLUSTRATED ALONG SMB2**

Wolfgang Grieskamp, Architect, Microsoft Corporation

Microsoft Protocol Engineering Team

- ◆ Fred Wurden
- ◆ Dave MacDonald
- ◆ Keith Stobie
- ◆ Chris Kaler
- ◆ Nico Kicillof
- ◆ Wolfgang Grieskamp
- ◆ ... and many more

Part 1: Background

Microsoft's Interoperability Initiative

Principles <http://www.microsoft.com/interop/principles>:

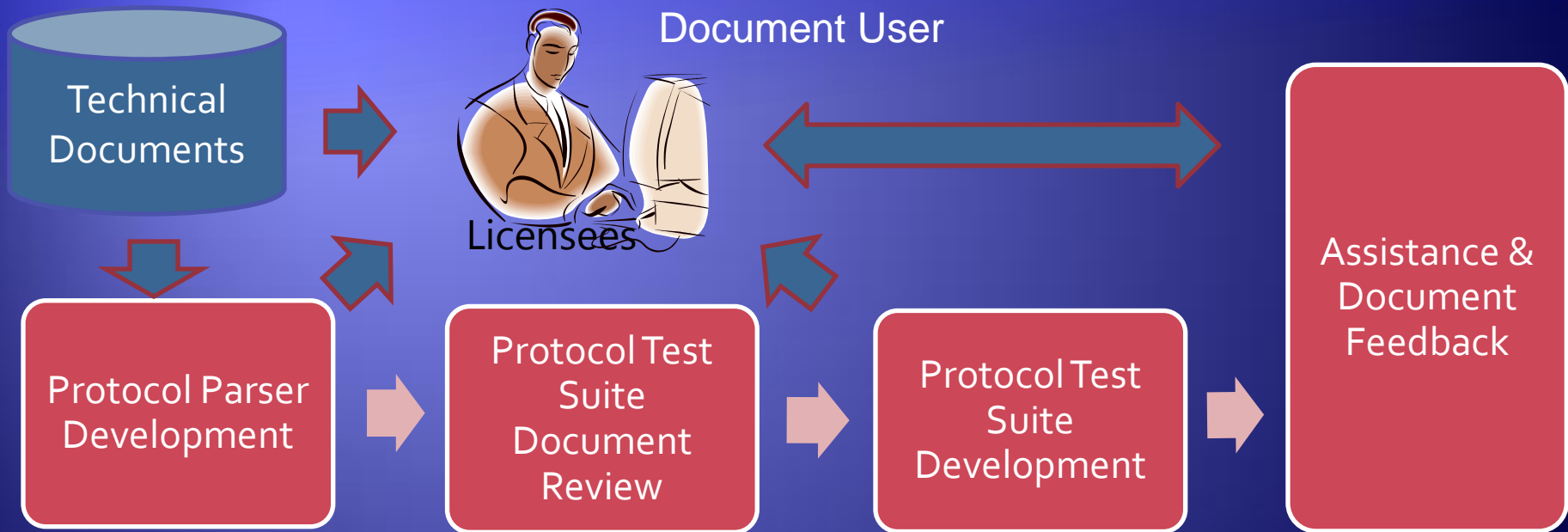
1. Open Connection to Microsoft Products
2. Support for Standards
3. Data Portability
4. Open Engagement

Microsoft's Interoperability Initiative

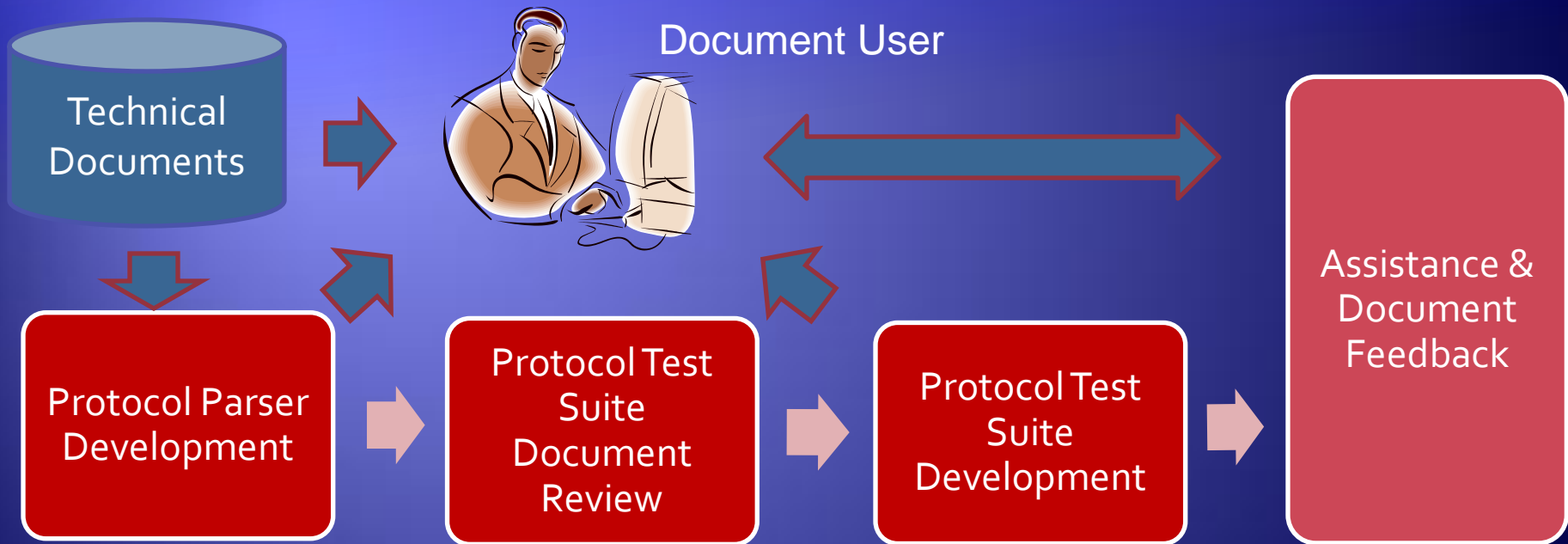
Principles <http://www.microsoft.com/interop/principles>:

1. Open Connection to Microsoft Products
 - ◆ Open Protocols <http://msdn2.microsoft.com/en-us/library/cc216514.aspx>
 - ◆ **Focus of this talk:**
Quality Assurance of Protocol Documentation
2. Support for Standards
3. Data Portability
4. Open Engagement

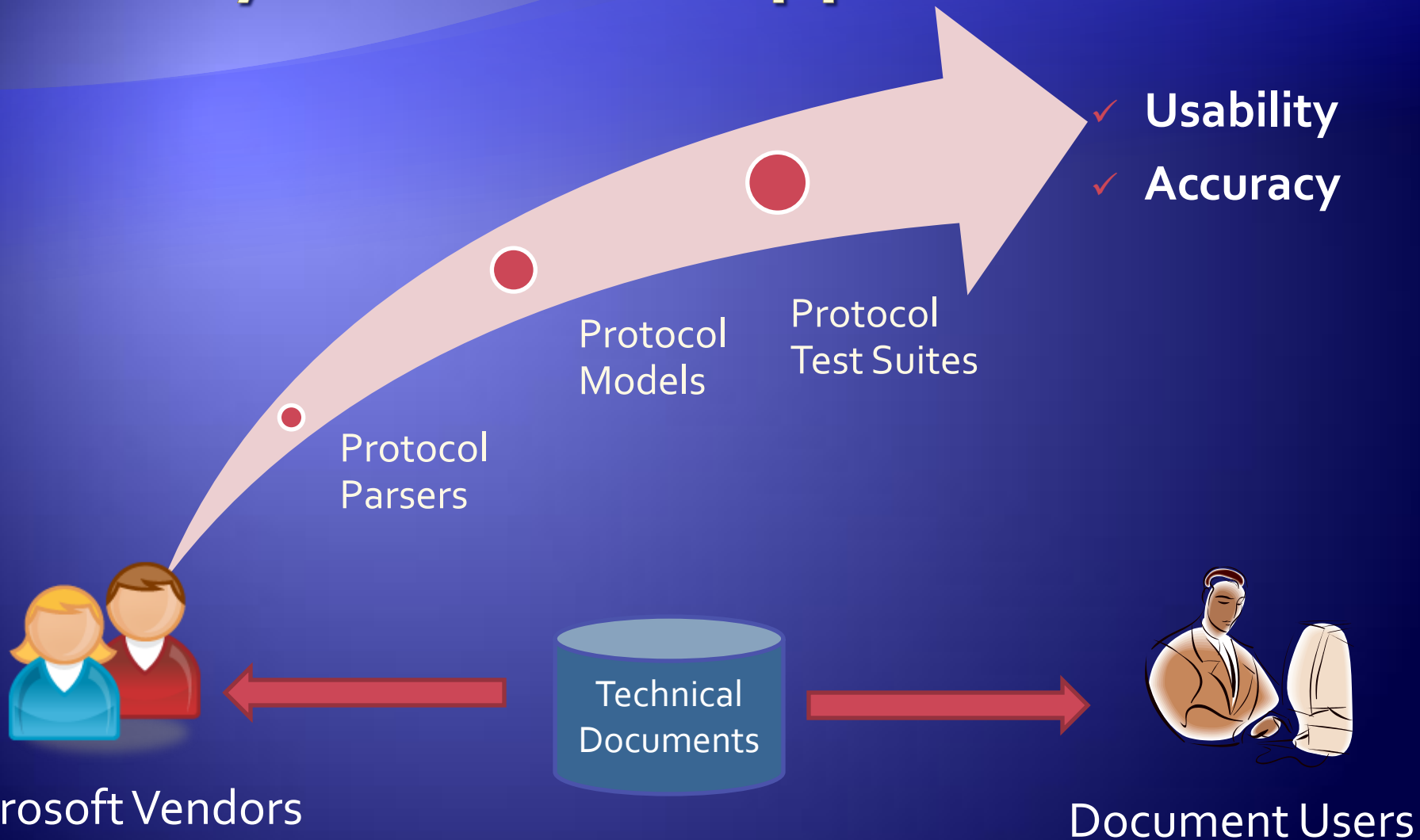
Microsoft's Technical Document Quality Assurance Efforts



Microsoft's Technical Document Quality Assurance Efforts



Document Quality Assurance Approach



Scope and Constraints

- ◆ In close sync with regulatory agencies in US and EU
- ◆ 250+ protocols in Windows alone (~ 30,000 pages of documentation)
 - ◆ Scope is extending (Office, .Net, ...)
- ◆ Clean-room approach
 - ◆ Vendors in India and China do test suite development
- ◆ Current investment
 - ◆ ~280 vendor employees for parser and test suite development
 - ◆ ~50 Microsoft employees for management and tools infrastructure

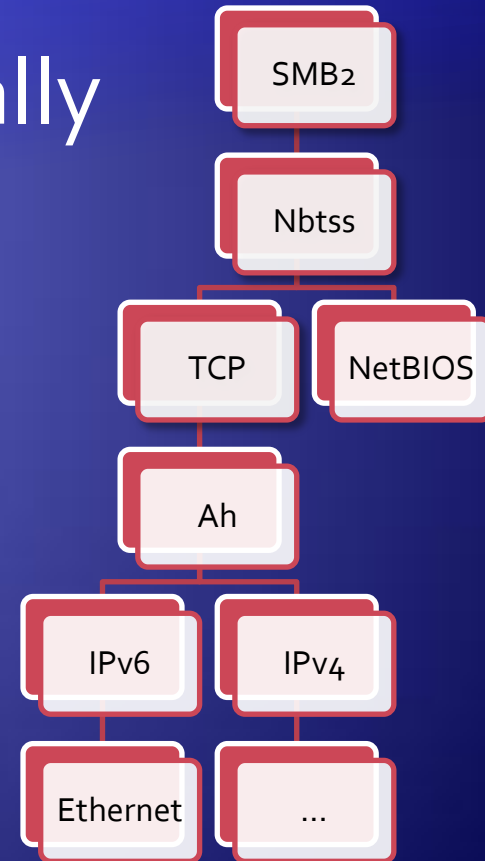
Part 2: Netmon and Protocol Parsers (in a nutshell)

Netmon: a network traffic analysis tool

- ◆ Can capture traffic on various interfaces
- ◆ Can do “almost real time” parsing
- ◆ Rich UI for analysis of traffic
- ◆ Based on declarative packet parser language
- ◆ Parsers will be moved to open source
 - ◆ MS-PL license

Netmon features

- ◆ Parsers defined hierarchically in script
 - ◆ Easily modified
 - ◆ Supports properties
 - ◆ Supports reassembly
 - ◆ Supports conversations
- ◆ Comes with SDK/API



Display Filter

History Verify Apply Remove SMB2

SMB 2

Display Filter Aliases

Frame Summary

Frame Number	Time Offset	Process Name	Conv Id	Source	Destination	Protocol Name	Description
2	0.015600			2001:4898:70...	SMB2TS.fareas...	SMB2	SMB2:C NEGOTIATE (0x0), GUID={00000000-0000-0000-0000-000000000000}, Mid = 0
3	0.031200			SMB2TS.fareas...	2001:4898:70...	SMB2	SMB2:R NEGOTIATE (0x0), GUID={4FE9CA2E-8D21-57AC-4431-FEB8857B8D62}, Mid = 0
40	0.280800			2001:4898:70...	SMB2TS.fareas...	SMB2	SMB2:C SESSION SETUP (0x1), Mid = 1
43	0.296400			SMB2TS.fareas...	2001:4898:70...	SMB2	SMB2:R SESSION SETUP (0x1), SessionFlags=0x0, Mid = 1
68	0.358800			2001:4898:70...	SMB2TS.fareas...	SMB2	SMB2:C TREE CONNECT (0x3), Path=\\SMB2TS\Share1, Mid = 2
69	0.358800			SMB2TS.fareas...	2001:4898:70...	SMB2	SMB2:R TREE CONNECT (0x3), TID=0x1, Mid = 2

Frame Details

Frame:

- Ethernet: Etype = IPv6, DestinationAddress: [00-18-8B-18-75-5A], SourceAddress: [00-07-B3-D2-ED-...]
- Ipv6: Next Protocol = AH, Payload Length = 284
- Ah: Next Protocol = TCP, SPI = 0x7444B74F, Seq = 0xB8C9
- Tcp: [Bad CheckSum]Flags=...AP..., SrcPort=Microsoft-DS(445), DstPort=56738, PayloadLen=240
- Nbtss: SESSION MESSAGE, Length =236
- Smb2: R NEGOTIATE (0x0), GUID={4FE9CA2E-8D21-57AC-4431-FEB8857B8D62}, Mid = 0
 - SMBIdentifier: SMB
 - Size: 64 (0x40)
 - Epoch: 0 (0x0)
 - Status: 0x0, Facility = FACILITY_SYSTEM, Severity = STATUS_SEVERITY_SUCCESS, Code = (0) SI
 - Command: NEGOTIATE (0x0)
 - Credits: 1 (0x1)
 - Flags: 0x1
 - NextCommand: 0 (0x0)
 - MessageId: 0 (0x0)
 - ProcessId: 4884 (0x1314)
 - TreeId: 0 (0x0)

Hex Details

Decode As Columns Prot Off: 4 (0x04)

0037	04	00	00	74	44	...	tD
003C	B7	4F	00	00	B8	.	O...
0041	C9	46	10	51	31	É	F.Q1
0046	BF	0E	F3	C5	0D	ç	ó.Å.
004B	28	B6	E2	01	BD	(á.º
0050	DD	A2	52	FF	00	Ý	çRÿ.
0055	55	E5	C5	C3	5F	U	â.Å.Å.
005A	50	18	01	03	96	P	...
005F	5D	00	00	00	00]
0064	00	EC	FE	53	4D	.	ipSM
0069	42	40	00	00	00	B	@...
006E	00	00	00	00	00
0073	00	01	00	01	00
0078	00	00	00	00	00
007D	00	00	00	00	00
0082	00	00	00	00	14
0087	13	00	00	00	00
008C	00	00	00	00	00
0091	00	00	00	00	00
0096	00	00	00	00	00

Part 3: Protocol Test Suite Development

Testing of *Documents*

Developing model and test suite

- ◆ From document alone
- ◆ By vendor without previous MS internal knowledge

Ensures:

- ✓ Usability of document
 - ◆ Simulates developer situation
- ✓ Accuracy of document
 - ◆ Discovers discrepancy between document and implementation

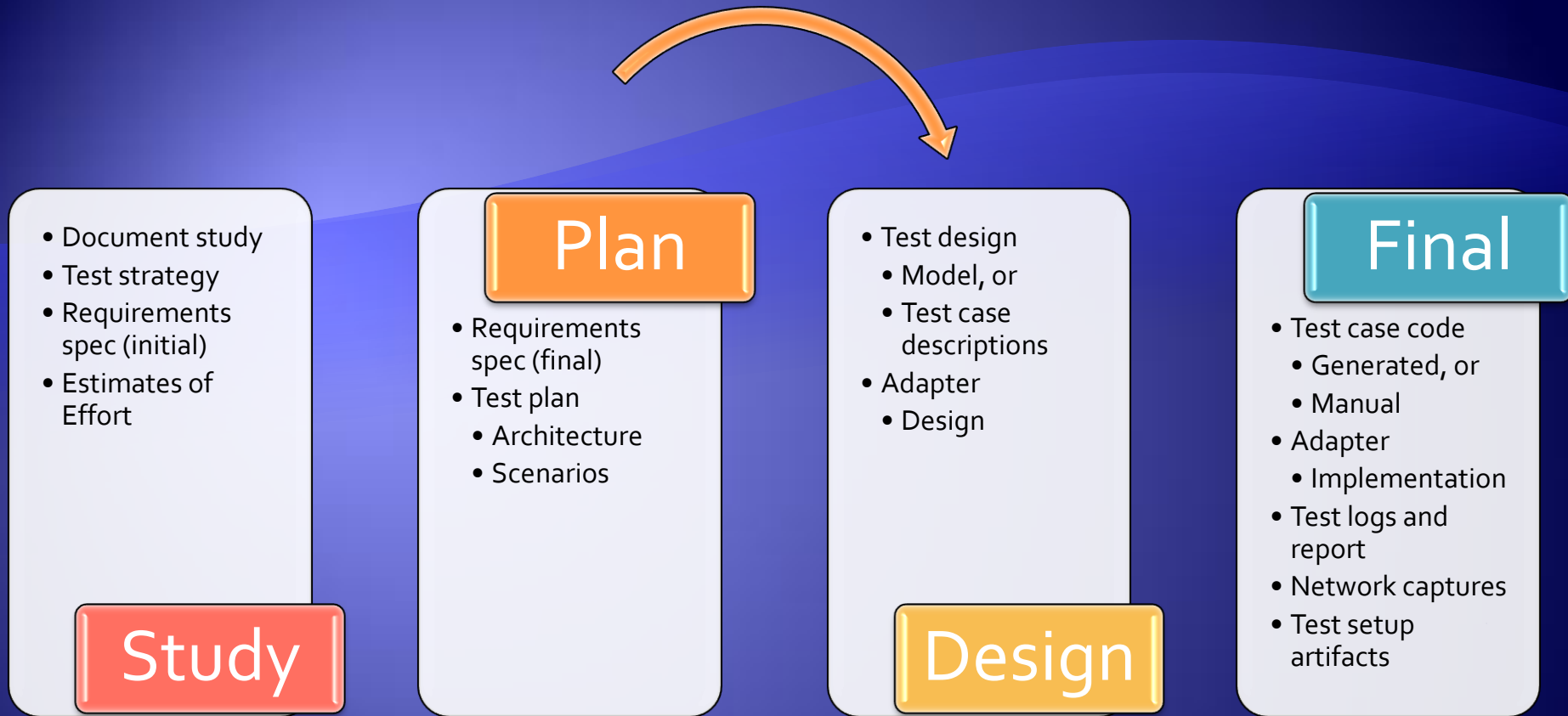
Out of Scope

- ◆ Exhaustive implementation testing
- ◆ Stress/performance testing
- ◆ Certification testing
- ◆ Client behavior testing

Protocol Quality Assurance Process (PQAP)



PQAP Deliverables



Protocol Quality Assurance Report (PQAR)

Requirement Gathering

- ◆ Predefined template and guidelines for
 - ◆ Identifying requirements in the spec
 - ◆ Classifying them according to
 - ◆ verifiability criteria
 - ◆ verification strategy (manual, model, adapter)
- ◆ Requirement gathered from spec alone
 - ◆ Gatherers are not expert in the particular protocol
 - ◆ Any ambiguity or clarity issue filed as TDI
 - ◆ Resulting Requirement Spec reviewed by the reviewing team
 - ◆ Windows-specific behavior listed as separate requirements
- ◆ Requirement Spec used as input for model design

Requirements Specification: SMB2

[MS-SMB2] Requirement Spec.xlsm [Read-Only] - Microsoft Excel

Home Insert Page Layout Formulas Data Review View

Clipboard Font Alignment Number Styles Cells Editing

A571 MS-SMB2-554

	A	B	C	D	E	F	G	H	I	J	L
1717	MS-SMB2-1700	3.3.5.15.7	When the server receives a request with an SMB2 header with a Command value equal to SMB2 IOCTL, and a CtlCode not listed above, if the operation succeeds, the server MUST then construct an SMB2 IOCTL response with the following values: CtlCode MUST be set to the CtlCode of the request.	S46	Non-extension	Protocol	Server p0		Normative	Test Case	
1718	MS-SMB2-1701	3.3.5.15.7	When the server receives a request with an SMB2 header with a Command value equal to SMB2 IOCTL, and a CtlCode not listed above, if the operation succeeds, the server MUST then construct an SMB2 IOCTL response with the following values: FileId MUST be set to Open.FileId.	S46	Non-extension	Protocol	Server p0		Normative	Test Case	
1719	MS-SMB2-1702	3.3.5.15.7	When the server receives a request with an SMB2 header with a Command value equal to SMB2 IOCTL, and a CtlCode not listed above, if the operation succeeds, the server MUST then construct an SMB2 IOCTL response with the following values: InputOffset MUST be set to the offset, in bytes, from the beginning of the SMB2 header to the Buffer[] field of the response.	S46	Non-extension	Protocol	Server p0		Normative	Adapter	
		3.3.5.15	When the server receives a request with an SMB2 header with a Command value equal to SMB2 IOCTL, and a CtlCode not listed above, if the operation succeeds, the server MUST then		Non-					Non-	Server interna

Requirements ScenarioReq Traceability Matrix Blocking Issues Usage

Ready 100%

Protocol Quality Assurance Report (PQAR)

- ◆ Template based document which is incrementally produced
- ◆ Central point of documentation of progress
- ◆ After finalization, turns into test suite documentation for sustained engineering

PQAR: SMB2

[MS-SMB2] Protocol Quality Assurance Report.doc (Read-Only) [Compatibility Mode] - Microsoft Word

Home Insert Page Layout References Mailings Review View

Print Layout Full Screen Reading Web Layout Outline Draft Document Views

Ruler Document Map Gridlines Thumbnails Message Bar Show/Hide

Zoom 100% One Page Two Pages Page Width


New Window Arrange All Remove Split View Side by Side Synchronous Scrolling Reset Window Position Window

Switch Windows Macros

Document Map

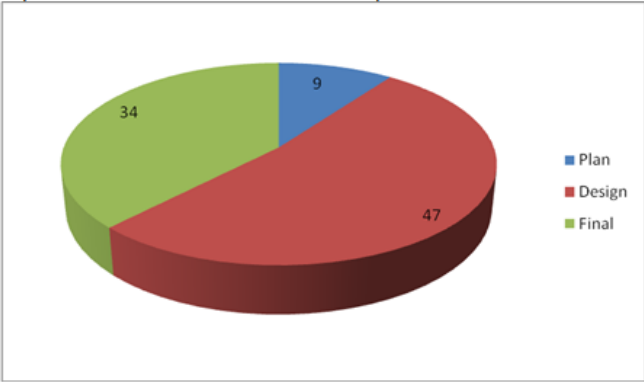
- 2.1.4 Constraints
- 2.2 Test Suite Approach
 - 2.2.1 System under Test (SUT)
 - 2.2.2 Abstraction
 - 2.2.3 Operation
 - 2.2.4 Actions
 - 2.2.5 Interface detail
- 2.3 Technical Feasibility of Message Gene
 - 2.3.1 Message Generation
 - 2.3.2 Adapter Approach
 - 2.3.3 Technical Issues
- 2.4 Dependencies/Considerations
- 2.5 Scenarios
- 2.6 Requirements coverage
- 2.7 TDIs filed
- Test Suite Design
 - 3.1 Model or Test Design Description
 - 3.1.1 Inner Working of Model
 - 3.1.2 Typical Scenarios
 - 3.2 Test Environment
 - 3.3 Test Preparation
 - 3.4 Test Cases
 - 3.5 Adapter Design
 - 3.5.1 Inter-working of Adapter, Test S
 - 3.5.2 Inner working of Adapter
 - 3.5.2.1 SMB2 Adapter & Traditional
 - 3.5.2.2 Server Configuration Adapt
 - 3.5.3 Miscellaneous issues
 - 3.6 Exceptions/Deviations
 - 3.7 TDIs filed
- Test Implementation and Execution
 - 4.1 Overall Test Suite Summary
 - 4.2 Test Report
 - 4.3 TDIs Filed
 - 4.4 Product Bugs Filed
 - 4.5 Requirement Coverage Statistics
 - 4.6 Sustaining Engineering
- Housekeeping
 - 5.1 Checklist
 - 5.2 Related Links
 - 5.3 Issues
 - 5.4 Cuts
 - 5.5 Change History

4.3 TDIs Filed

 SMB2_TDI_Final.psq

SMB2 team at ATC filed 90 TDIs totally, 71 are closed and 19 are active. Note that other teams outside ATC also contributed to file TDIs against SMB2.

The chart below depicts SMB2 TDIs filed in different phases.



Phase	Count
Plan	9
Design	47
Final	34

Protocol Tools and Test Team Microsoft Confidential 69

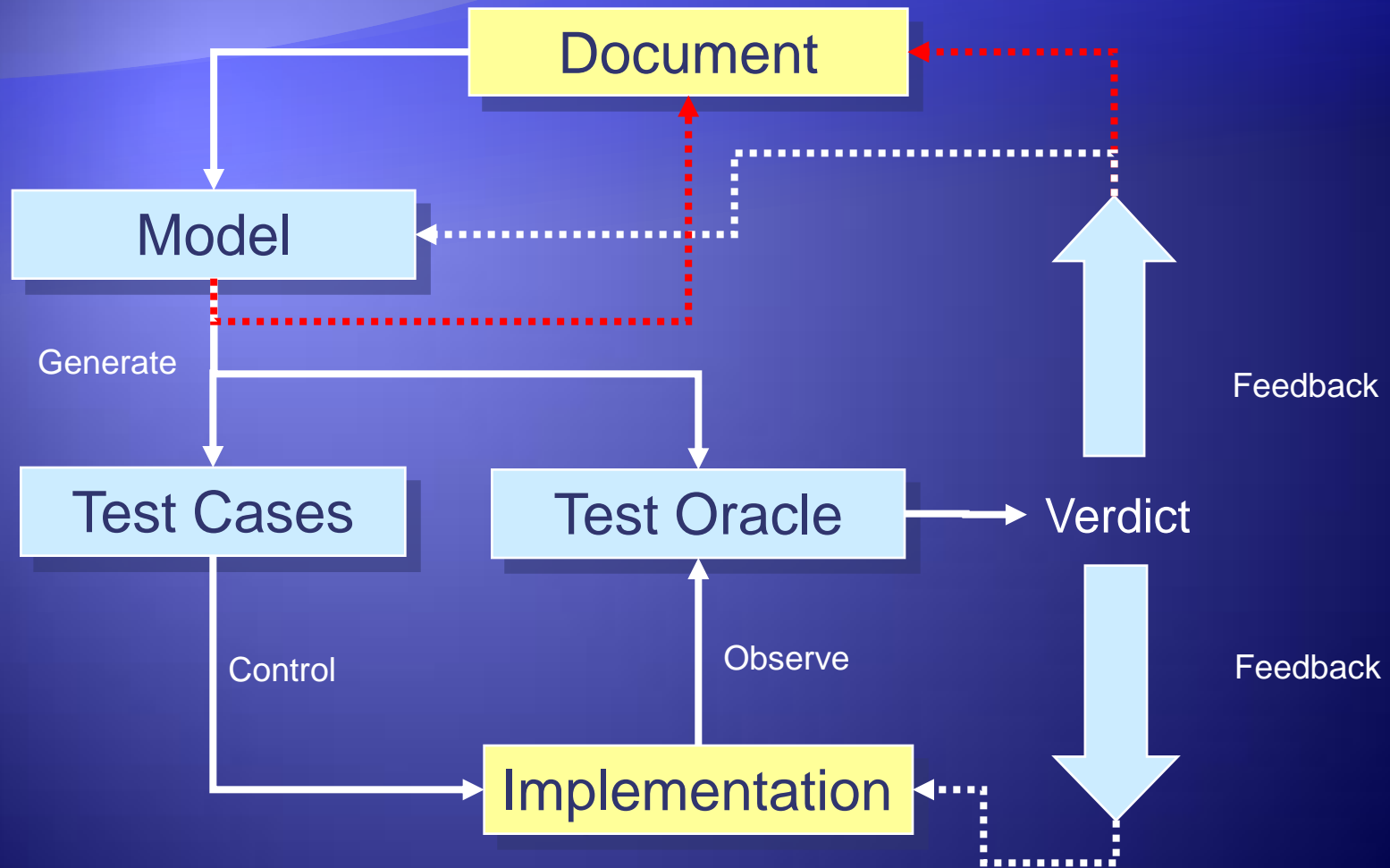
Page: 69 of 73 Words: 19,246 Recovered 120%

PQAP Review Process

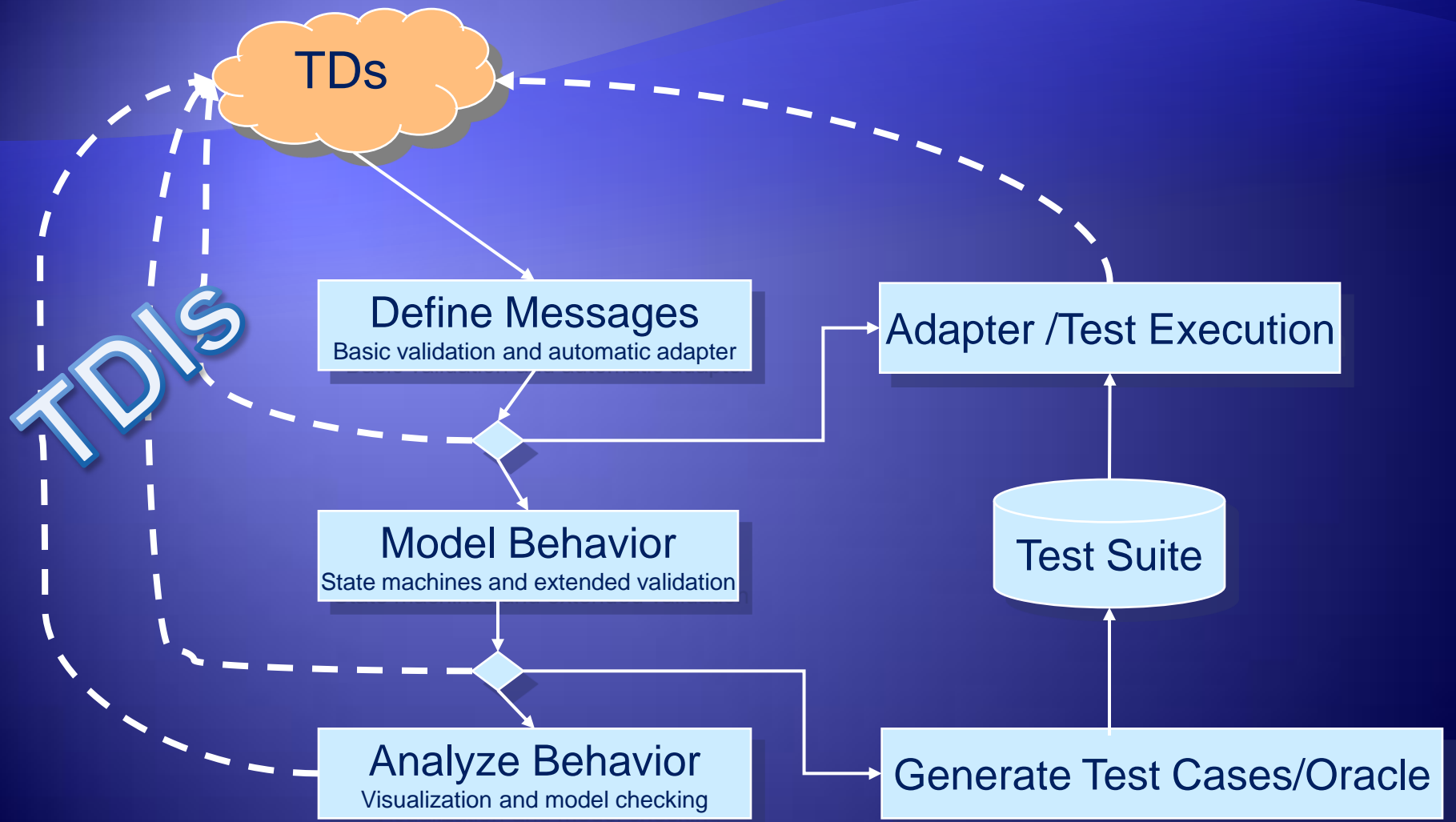
- ◆ Certified reviewers sign-off on phases of the PQAP
 - ◆ Internal and external industry experts which act independently
 - ◆ Formalized certification process via apprenticeship model with existing CRs
- ◆ Dispositions: re-review, conditional (after changes), accept
 - ◆ Quality not deadline oriented
 - ◆ Quality bar contains measurements like initial state of document, estimated relevance for document users, etc.

Part 4: Model-Based Testing

Test Suite Development with MBT



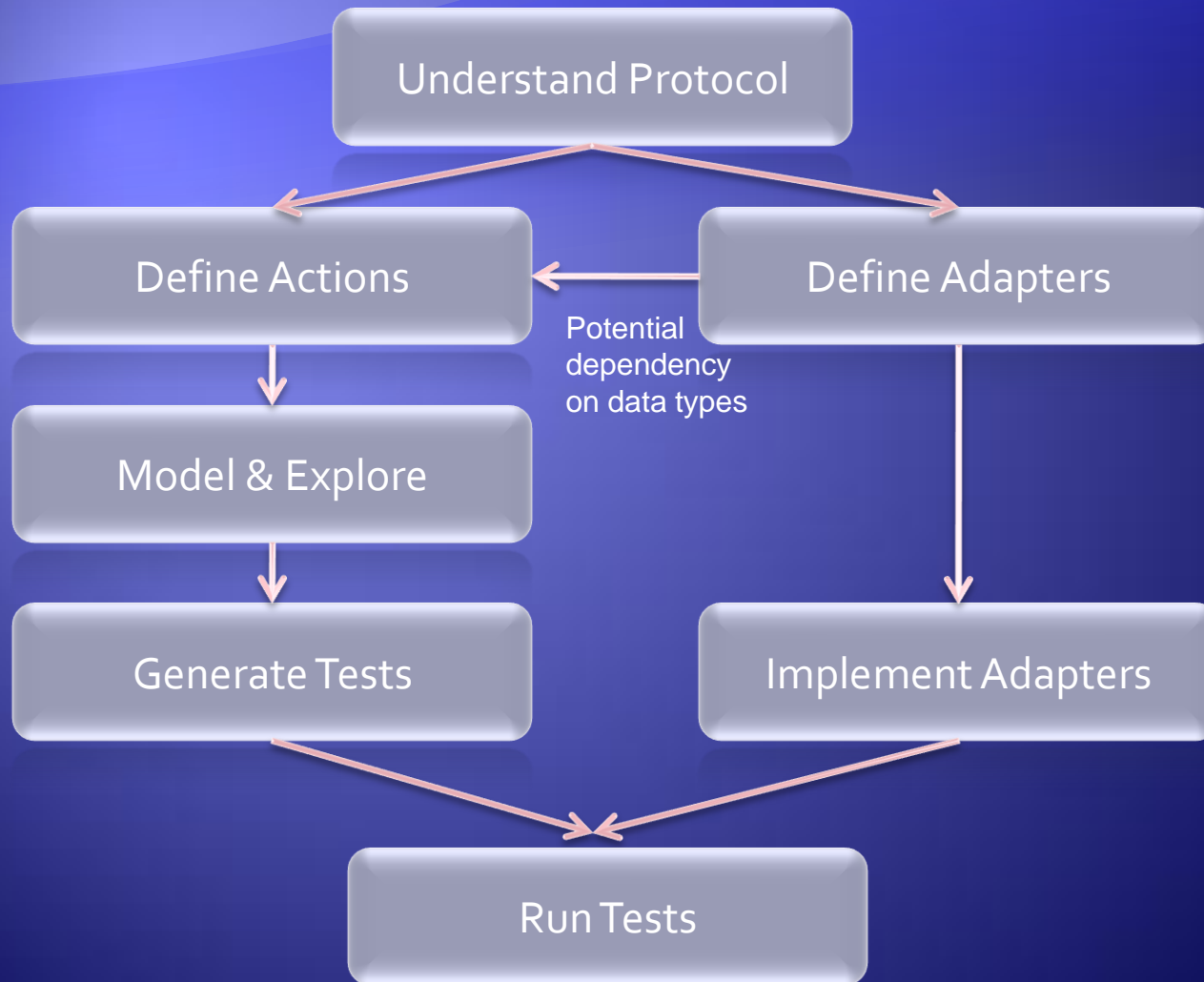
MBT Process for Protocols



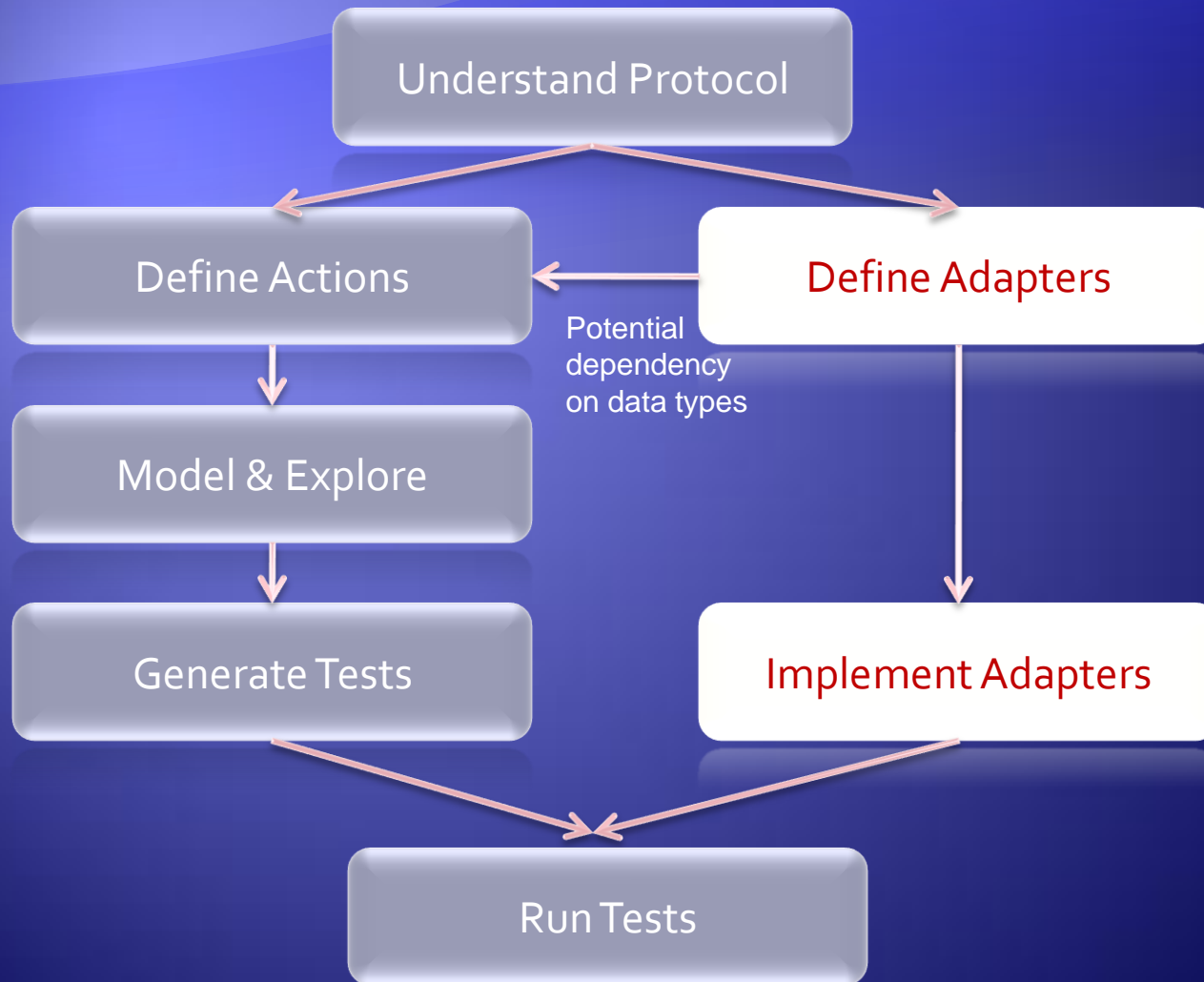
MBT Technology: Spec Explorer 2007

- ◆ Matured technology from Microsoft Research (first version 2002)
- ◆ Multiple modeling styles and languages (programs, patterns, diagrams)
- ◆ Extraction of state machine from infinite model
- ◆ Generation of test code from state machine
- ◆ Model composition
- ◆ Integration into Visual Studio

Model-Based Test Suite Development Process Drilldown

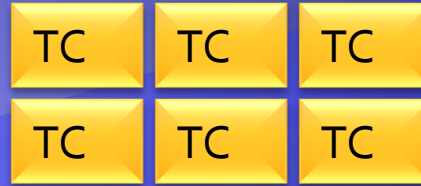


Model-Based Test Suite Development Process Drilldown



Test Adapters

Test Cases
(traditional or MBT)



- Abstracts SUT functionality
- Contract between teams
 - Test case team
 - Adapter team

Test Development

Adapter Interface

Adapter Development

Adapter Implementation



SUT

(System under test)

- Pluggable
 - Different server setups
 - Different transports
- Choice of adapter implementation flavors
 - Interactive (automatic)
 - Script (set of commands)
 - Managed (.NET code)
 - RPC (automatic)

Adapter interface: SMB2

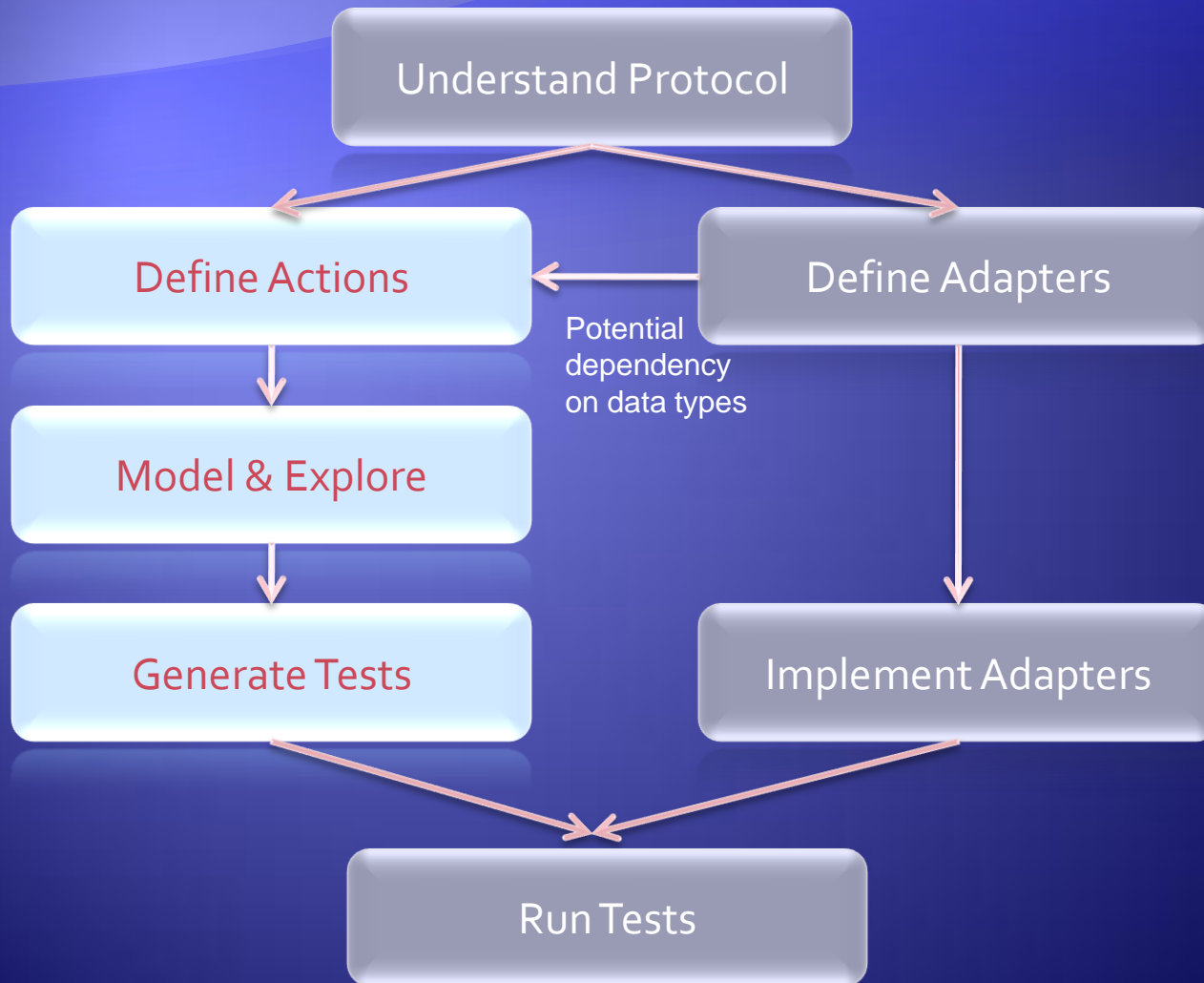
```
public interface ISmb2SetupAdapter : IAdapter
{
    void AssumeShareExists(int shareId, ShareType type);
    void AssumeShareDoesNotExist(int shareId);
    ...
}

public interface ISmb2Adapter : IAdapter
{
    void TreeConnectRequest(
        int relativeMessageId,
        int creditRequest, int shareId);
    event TreeConnectResponseHandler TreeConnectResponse;
    ...
}

public delegate void TreeConnectResponseHandler(int relativeMessageId,
    int creditResponse, int treeId, ShareType shareType);
```

- ◆ Methods represent *test control*
- ◆ Events represent *test observation*

Model-Based Test Suite Development Process Drilldown



Contract Model: SMB2

```
...
static SetContainer<int> fileIds;
static int fileIdsInFlight;
static SequenceContainer<Request> inflight;
...

[Action]
static void CreateRequest(int sequenceId, int creditRequest,
                        [Domain("OpenTreeDomain")]int treeId,
                        CreateType disposition,
                        [Domain("FileNameDomain")]string fileName)
{
    Contracts.Requires(fileIds.Count - fileIdsInFlight > 0);
    CheckRequest(sequenceId, creditRequest);
    inflight.Add(new CreateRequest(sequenceId, treeId, disposition, fileName));
    fileIdsInFlight++;
}
```

- ◆ Uses rich (infinite) model state
- ◆ Exploration slices an FSM

Test Selection: SMB2

```
machine StateMachine() : Actions
{
    construct model program from Actions where namespace = "SMB2.Model"
        // construct contract model from C#
}

machine AllSync() : Actions
{
    // compose contract model with test purpose
    (
        AssumeShareExists(1, ShareType.DISK);           // assume one share
        SetupConnectionAndSession(1);                   // setup session (window=1)
        ...                                              // wildcard from here
    )
    || StateMachine
}

machine TestsForAllSync() : Actions
{
    // construct test cases
    construct test cases where strategy = "longtests" for AllSync
}
```

Spec Explorer Demo

Config.cord [design] Smb2Adapter.cs Model.cs

Machine

- StateMachine
- SetupScenario
- SlicedStateMachine
- TestSuite
- <new machine>

Set as main Remove

Uses Configs

Actions

Behavior

(SetupScenario(); ...) || StateMachine

Validate

Explore (Incrementally)

Test Generate tests

Config

Actions

<new config>

Switches Types Actions Constraints Exceptions

Exploration

- BoundPath inherited
- BoundStates inherited
- BoundSteps 1024
- CodeGenerationTimeout inherited
- DepthFirst inherited
- Explorer inherited
- StepsUntilSuspend inherited
- StopAtError inherited

Solver

Solver

Testing

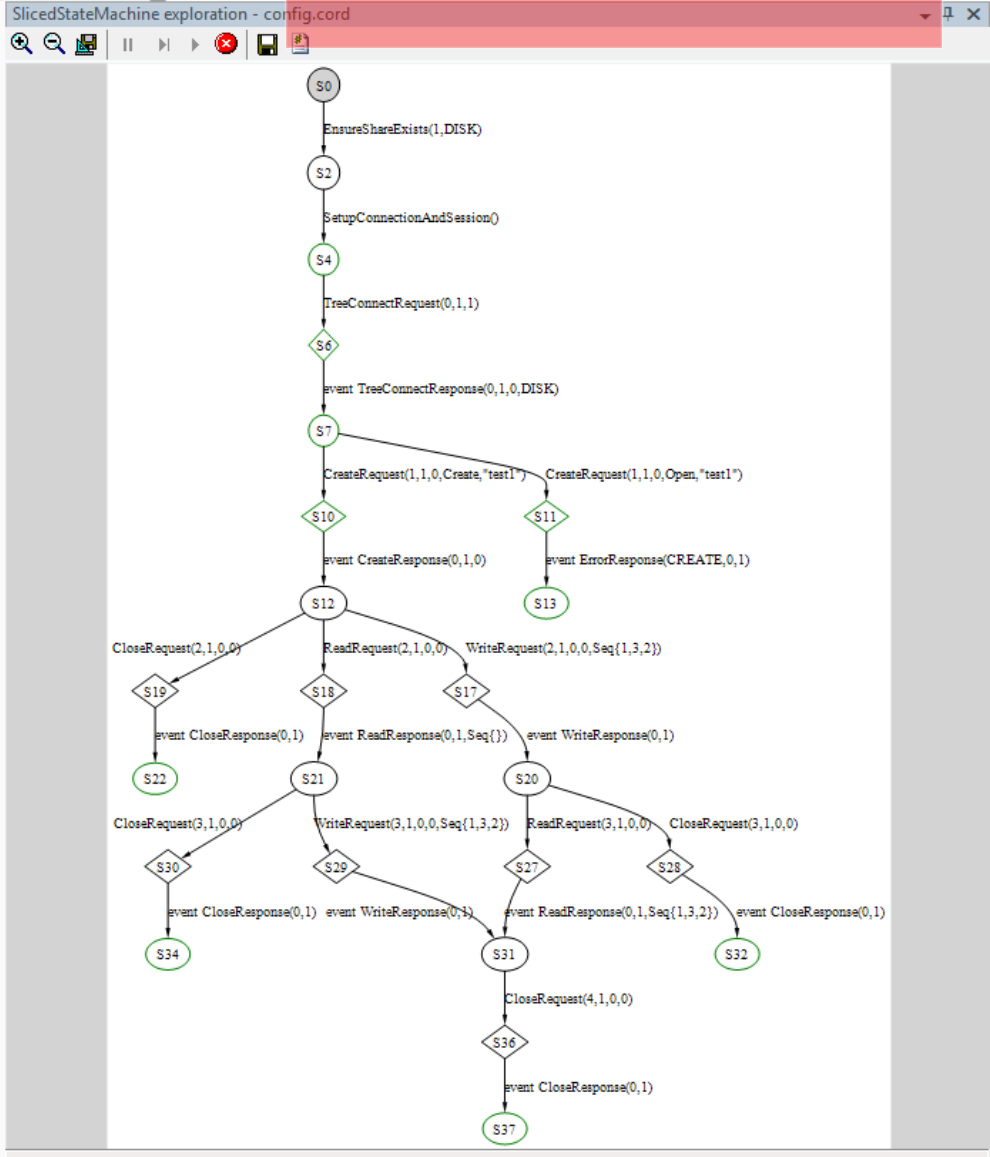
- ObservationBound inherited
- ProceedControlTimeout inherited
- QuiescenceTimeout inherited
- ReRuns inherited

Viewing

- CollapseLabels inherited
- CollapseSteps inherited
- DisplayRequirements inherited
- GraphTimeout inherited
- HideParameters inherited
- MaxLabelsPerArc 16
- NoViewer inherited
- ShowErrorsOnly inherited

BoundSteps

Defines a bound on the total number of steps to explorer.



Finished 36 states, 37 steps, 0 errors 00:00:01.822000

Part 4: Conclusions

Summary

- ◆ Comprehensive measurements to ensure document quality
 - ◆ But: testing can only prove the presence of errors!
- ◆ Using advanced technologies and processes
 - ◆ Driving the state of the art in the area
 - ◆ Proving that MBT scales in industry testing
- ◆ Making technologies available to the community
 - ◆ Many papers published
 - ◆ Netmon freely released soon, Spec Explorer to followup

The End (Thanks!)

Q&A