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:

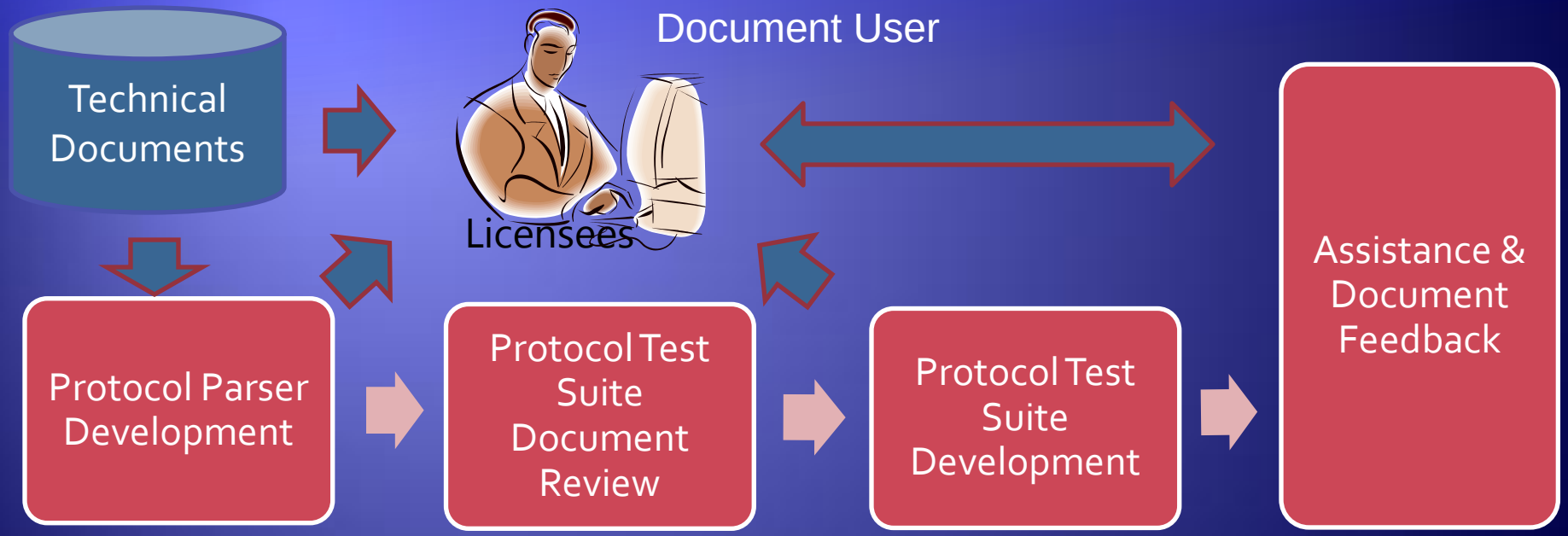
1. Open Connection to Microsoft Products
2. Support for Standards
3. Data Portability
4. Open Engagement

Microsoft's Interoperability Initiative

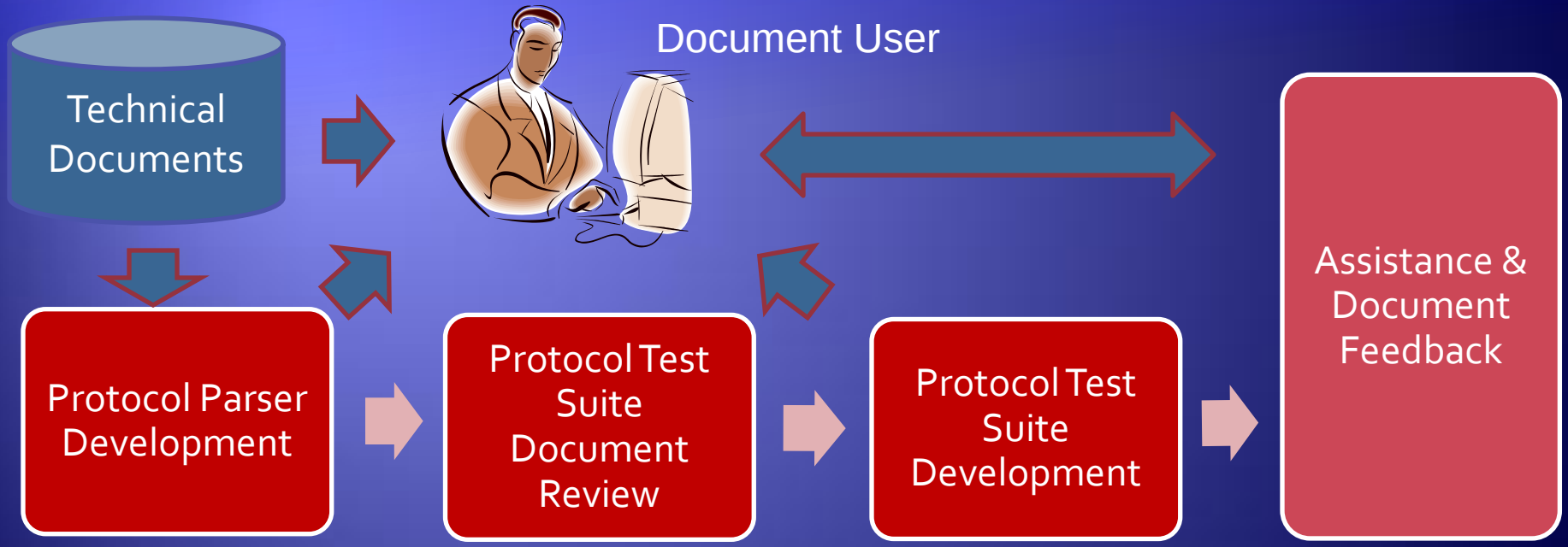
Principles http://www.microsoft.com/interop/principles:

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

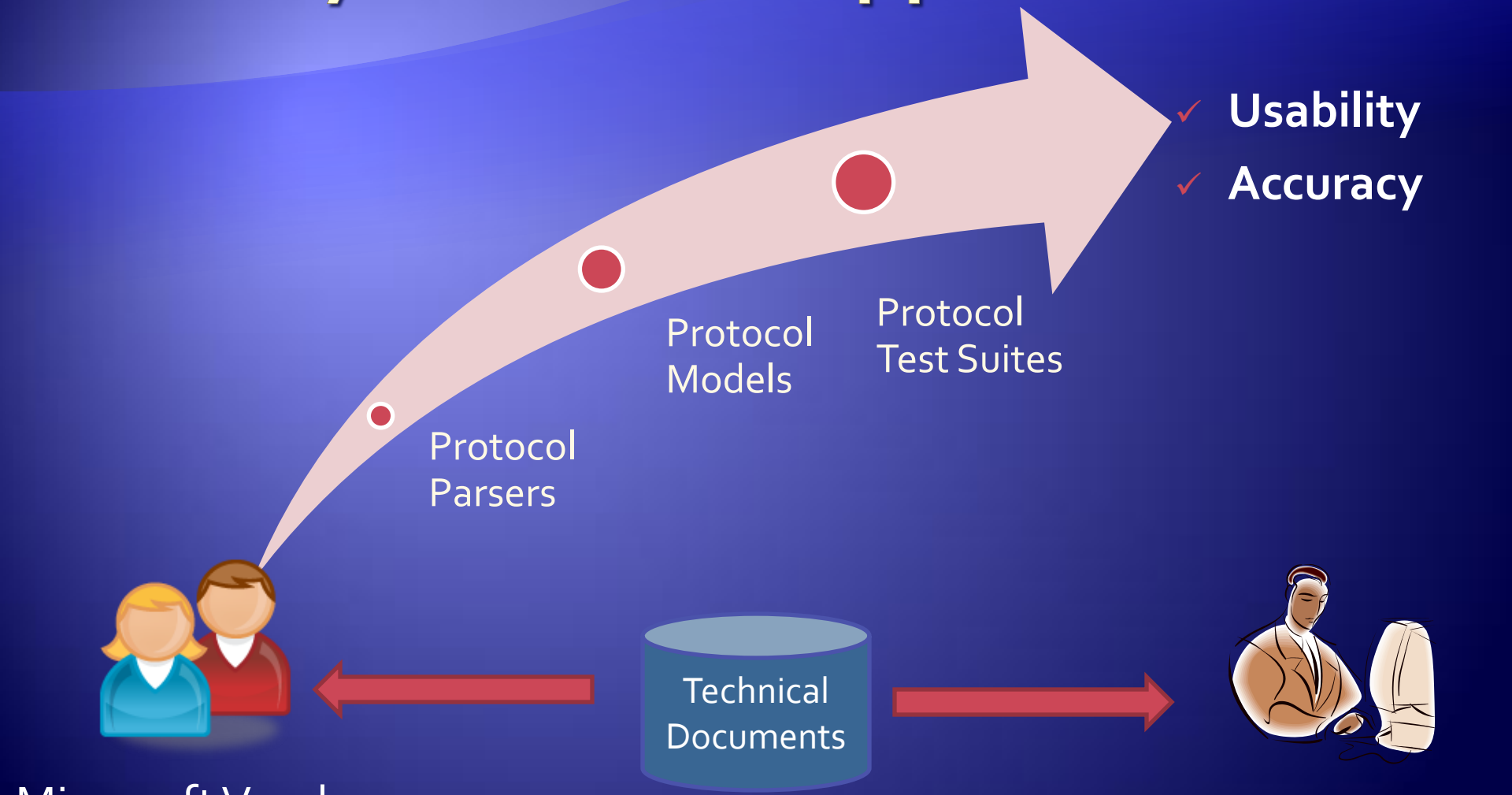
Microsoft's Technical Document Quality Assurance Efforts



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

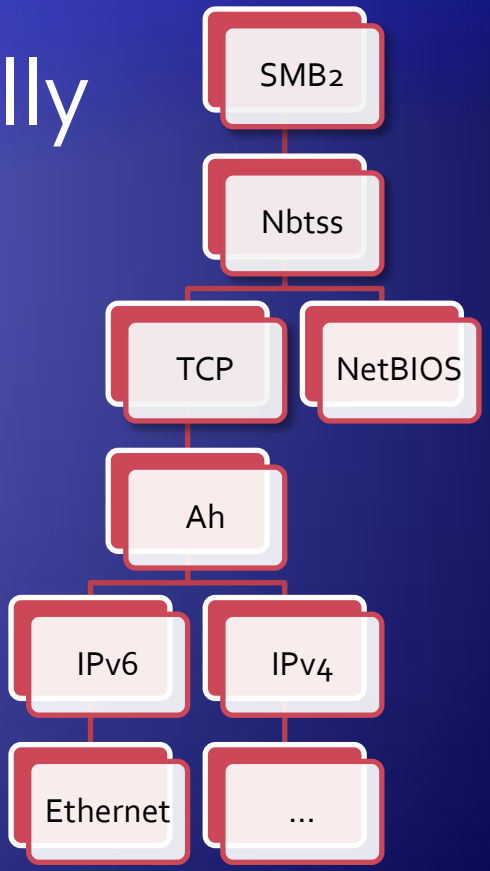
Part 2: Netmon and Protocol Parsers
(in a nutshell)

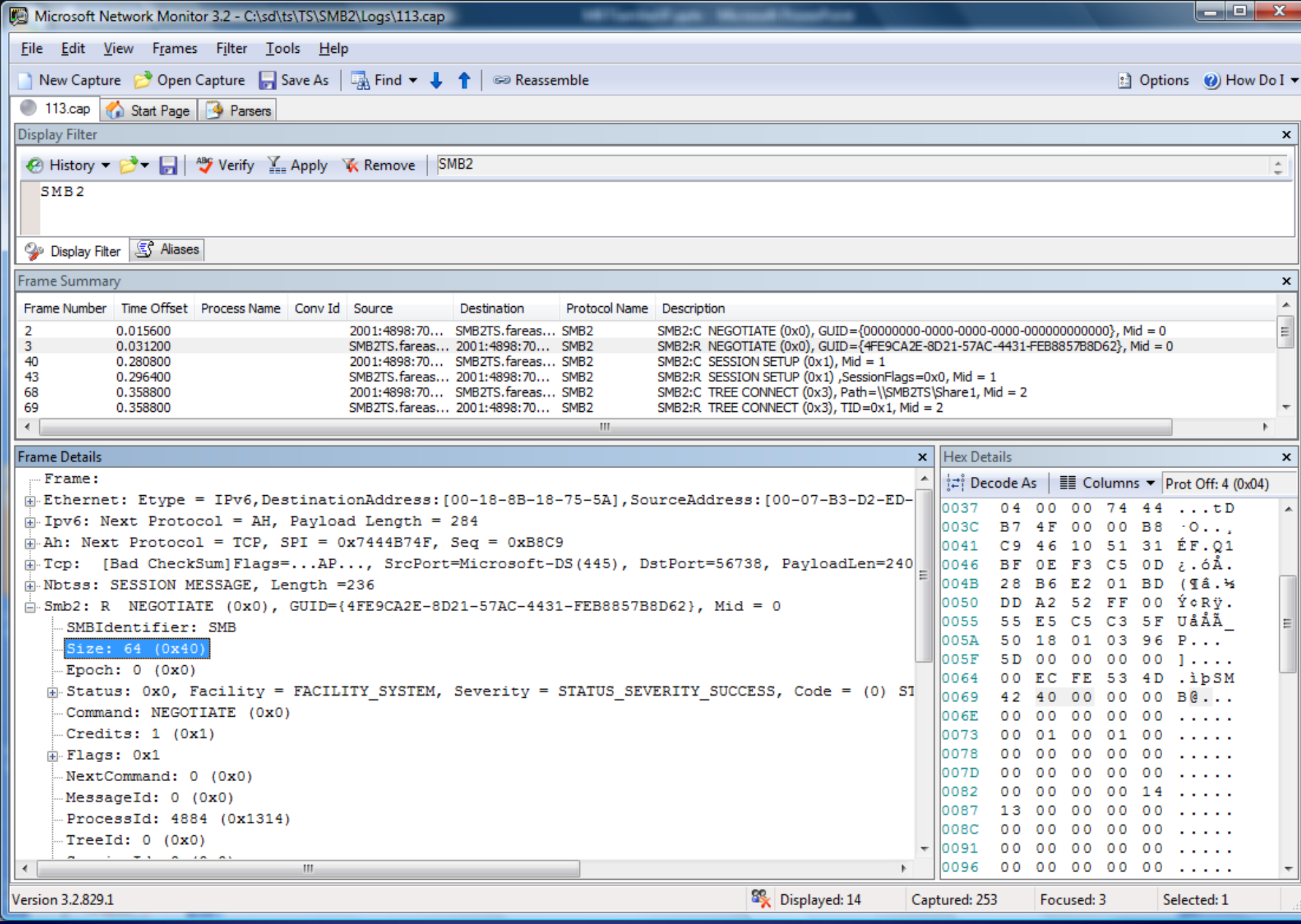
Netmon: a network traffic analysis tool

- 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





Part 3: Protocol Test Suite Development

Testing of *Documents*

Developing model and test suite

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

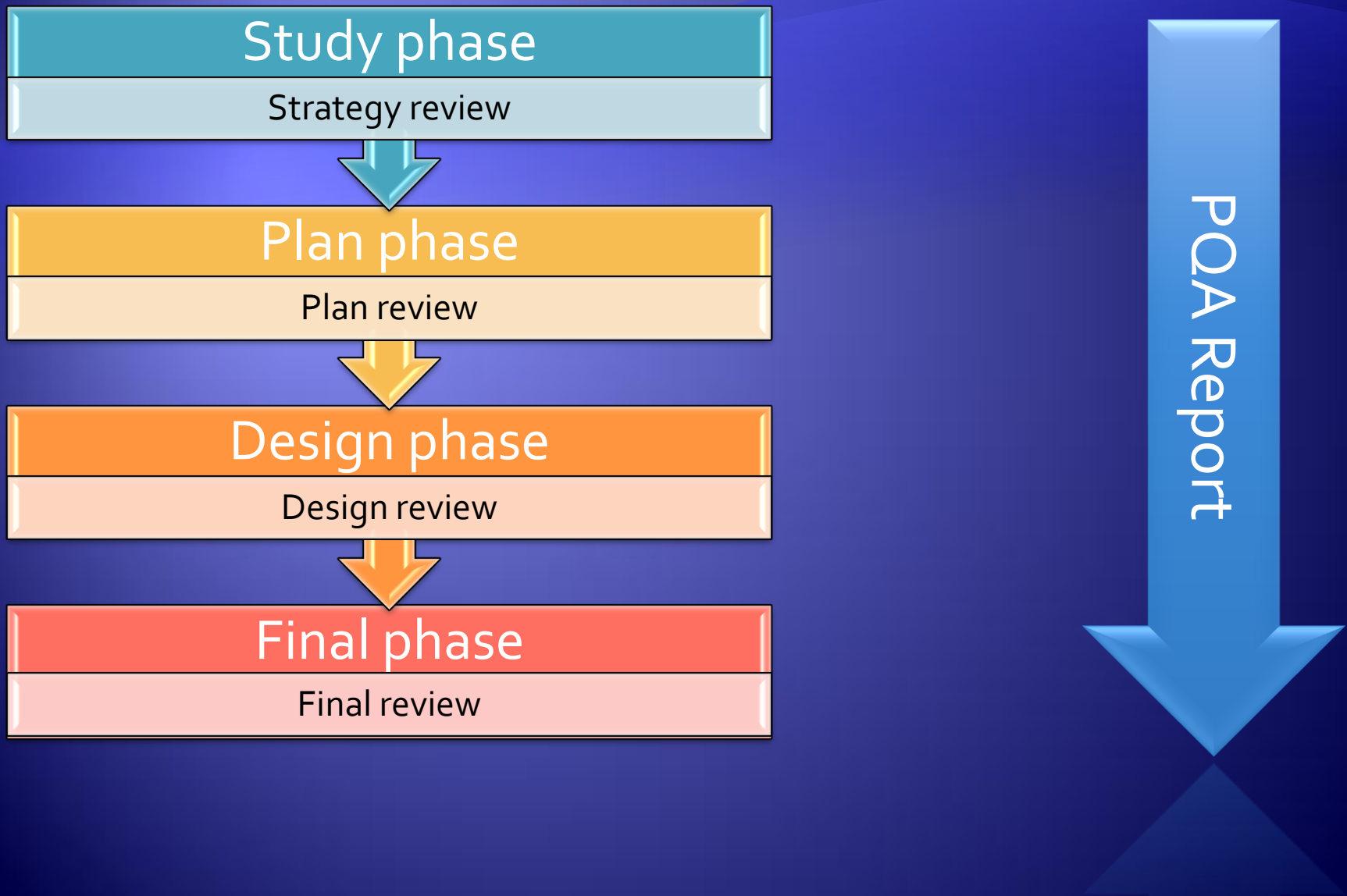
Ensures:

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

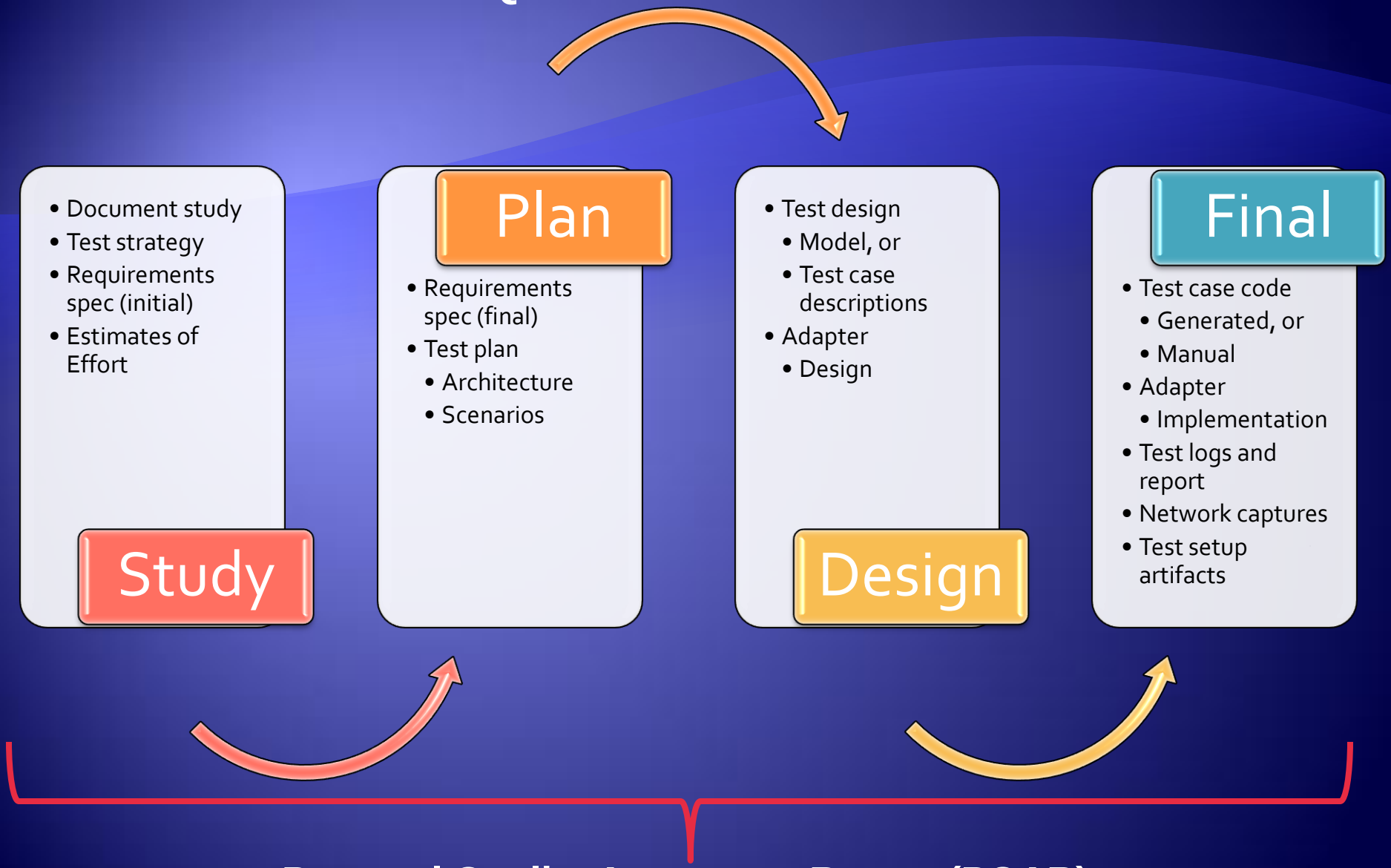
Out of Scope

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

Protocol Quality Assurance Process (PQAP)



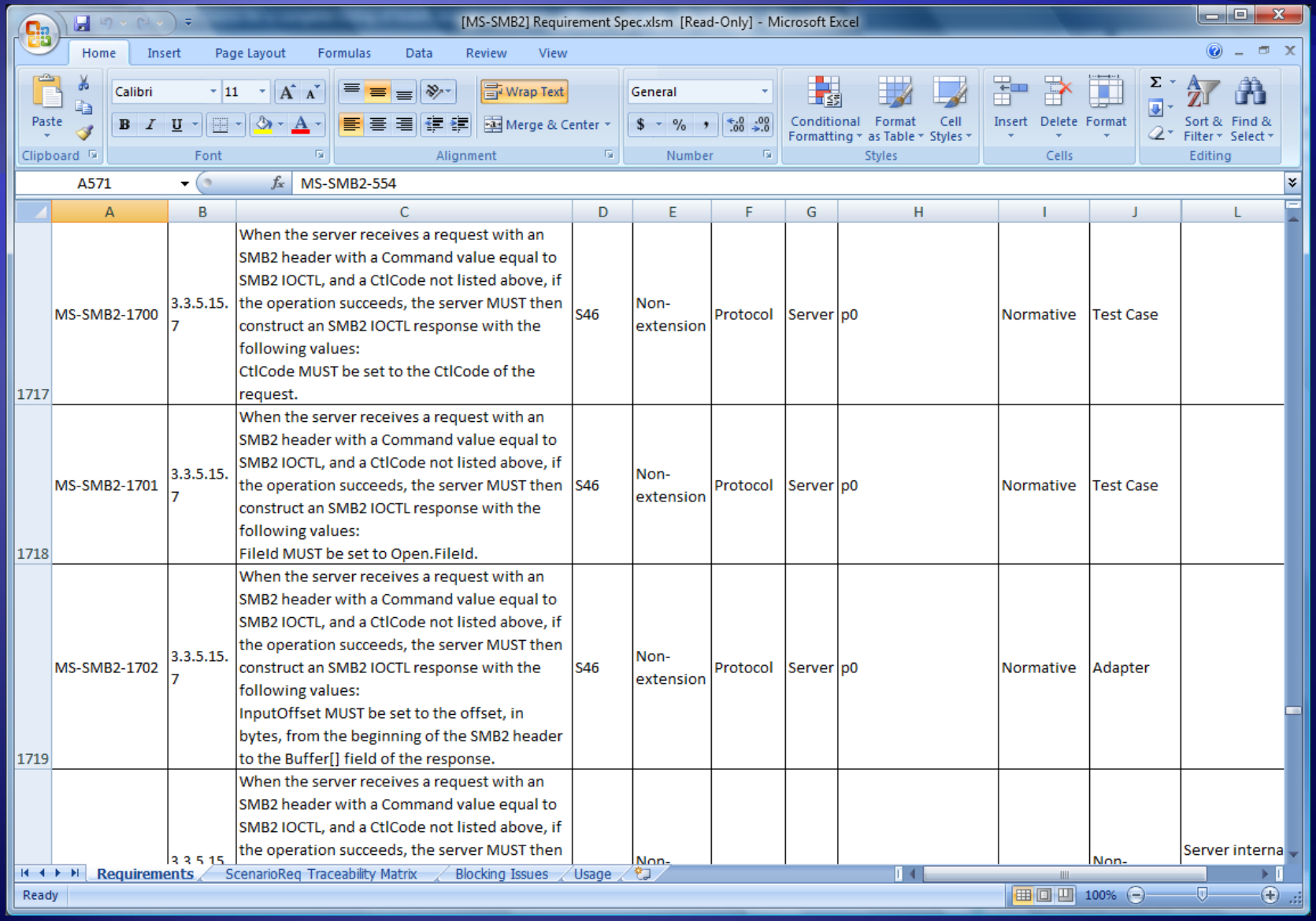
PQAP Deliverables



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



	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

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 | Gridlines | Message Bar | Document Map | Thumbnails — Show/Hide
Zoom | 100% | One Page | Two Pages | Page Width — Zoom
New Window | Arrange All | Remove Split | View Side by Side | Synchronous Scrolling | Reset Window Position | Switch Windows — Window
Macros — Macros

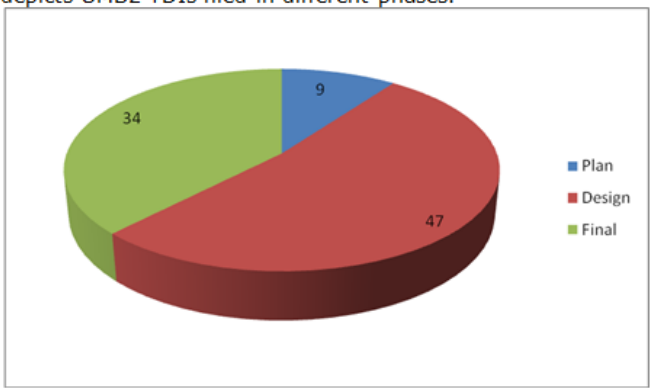
Document Map

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.



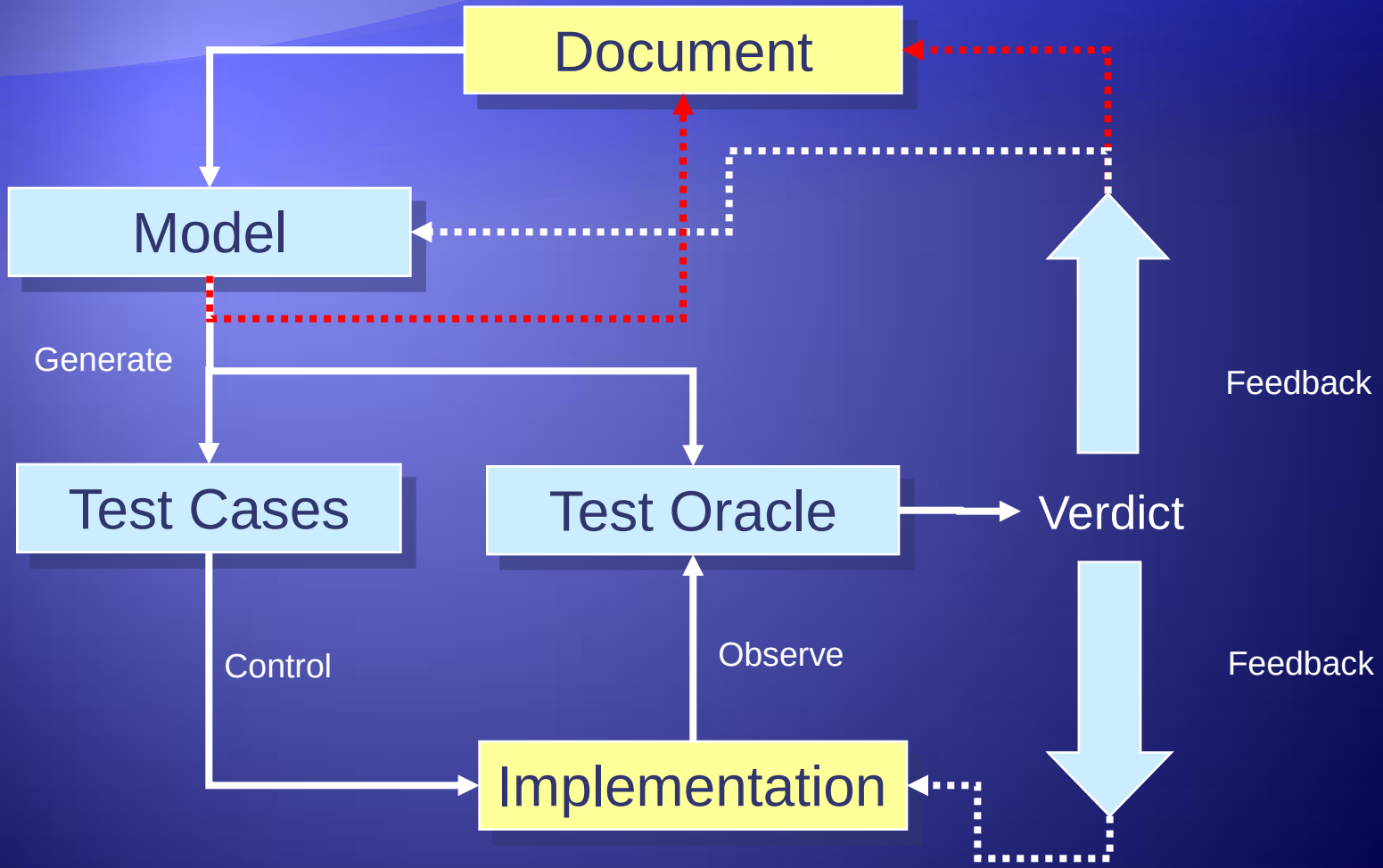
Page: 69 of 73 | Words: 19,246 | Recovered | 120%

PQAP Review Process

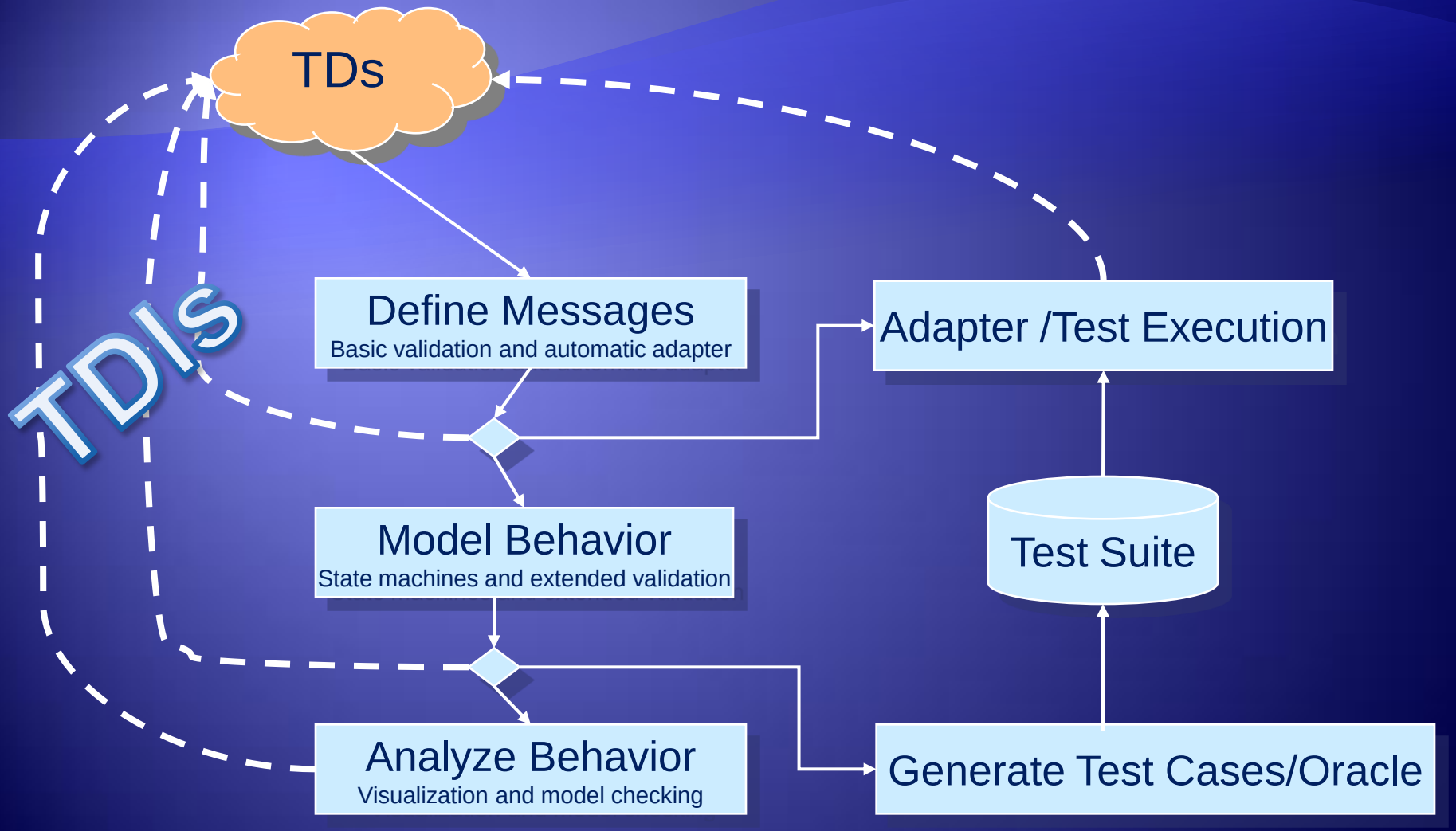
- Certified reviewers sign-off on phases of the PQAP
  - Internal and external industry experts which act independently
  - Formalized certification process via apprenticeship model with existing CRs
- Dispositions: re-review, conditional (after changes), accept
  - Quality not deadline oriented
  - 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



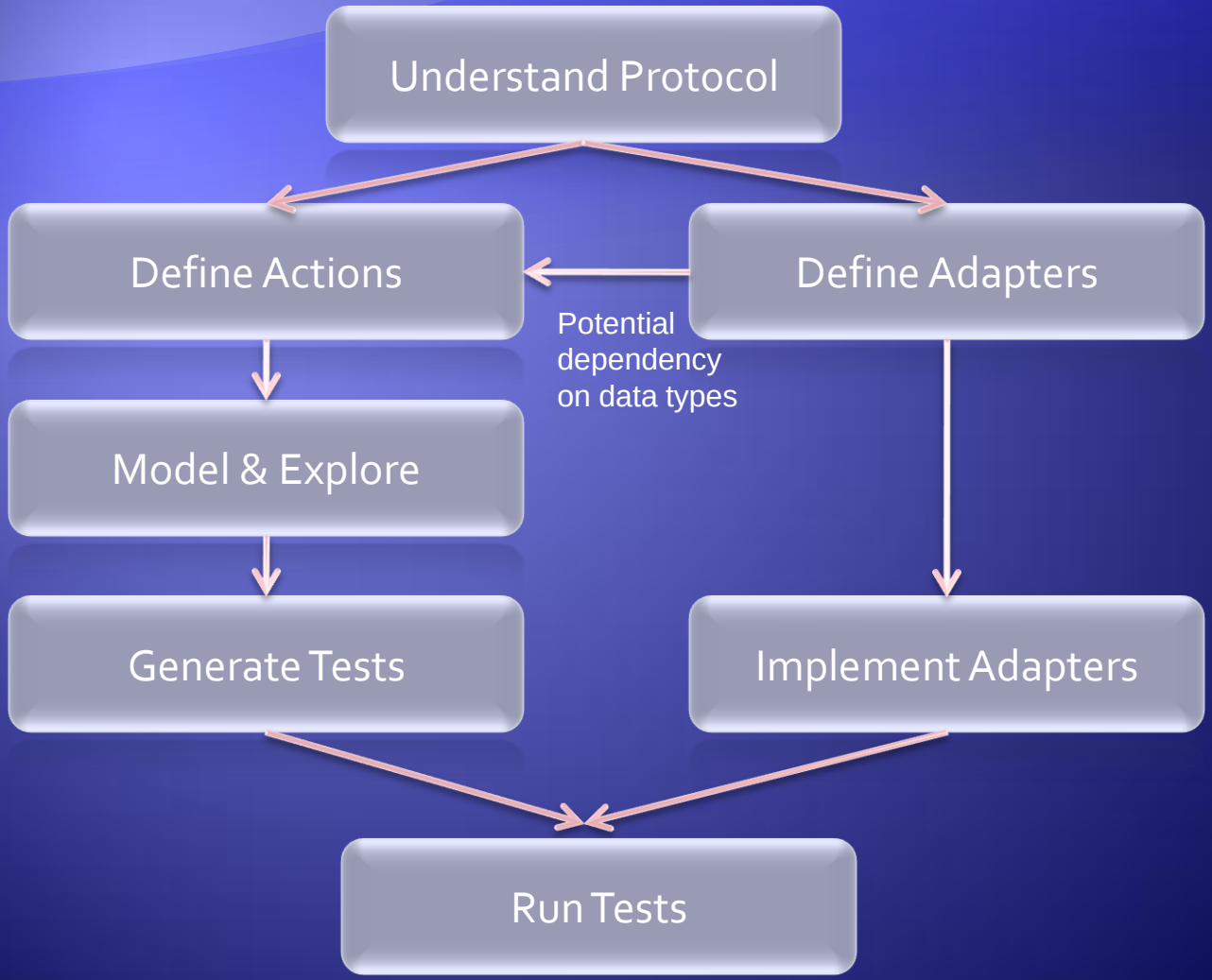
MBT Process for Protocols



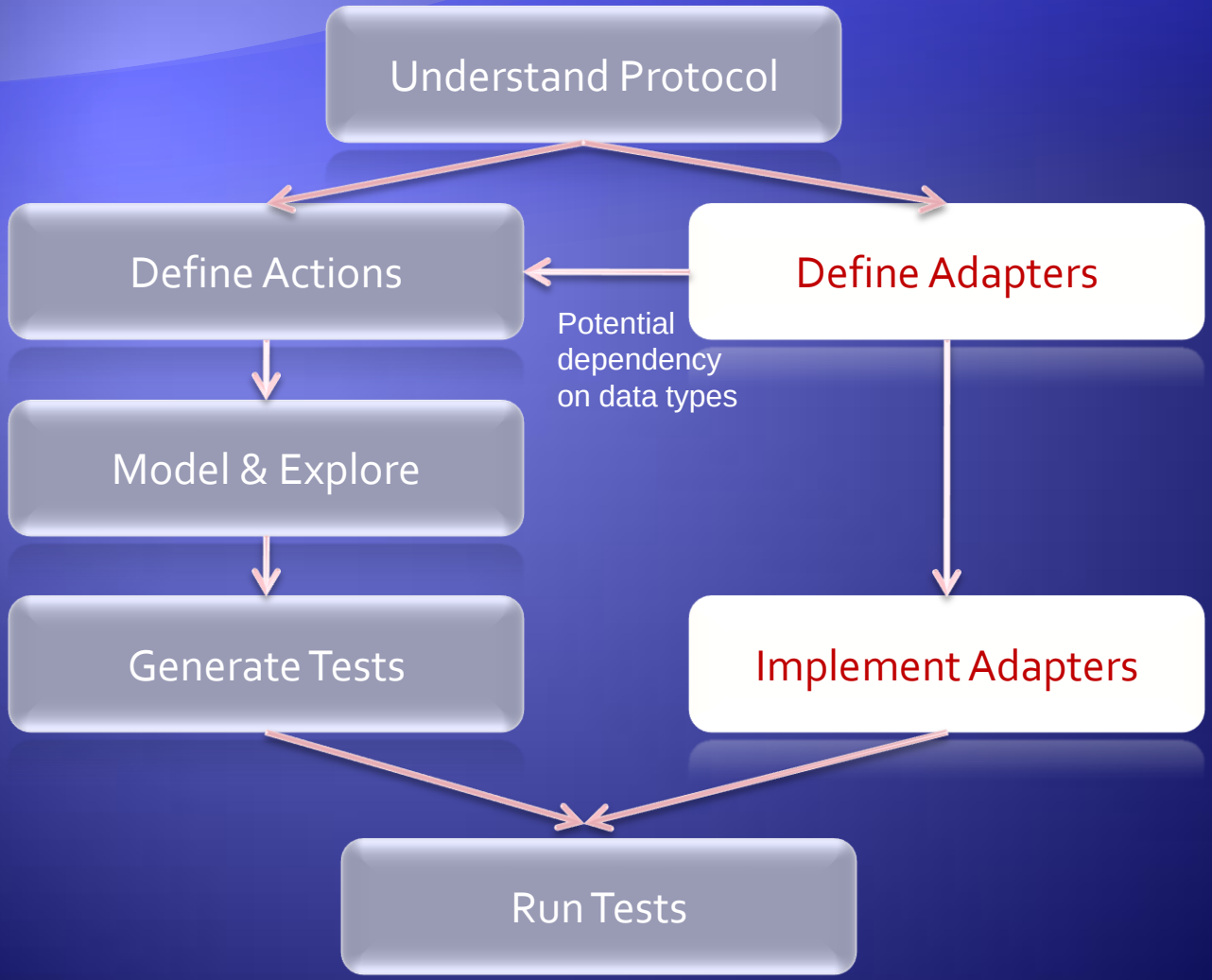
MBT Technology: Spec Explorer 2007

- 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)

TC TC TC
TC TC TC

Test Development

Adapter Interface

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

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

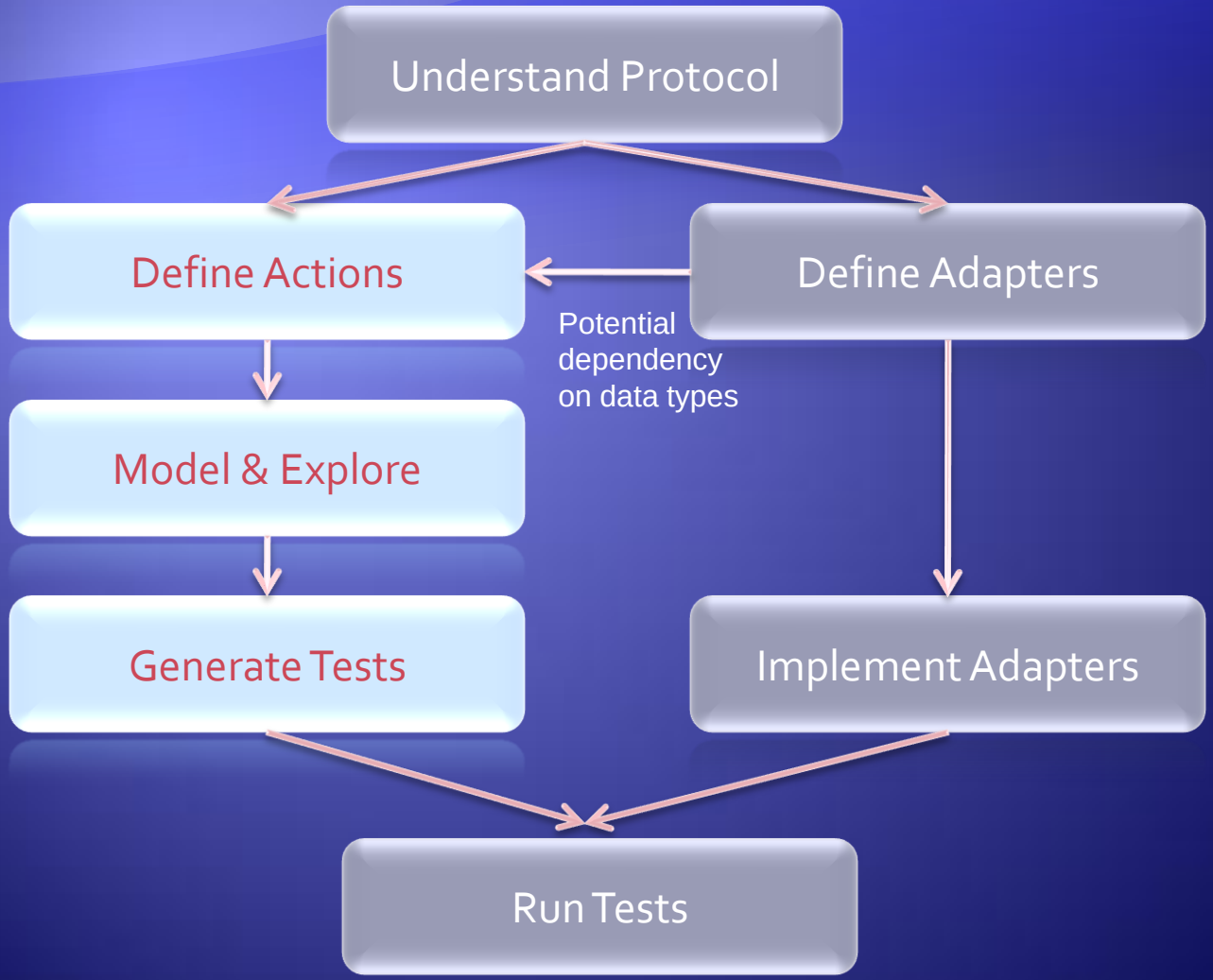
```
public interface ISmb2SetupAdapter : IAdapter
 {
     void AssumeShareExists(int shareId, ShareType type);
     void AssumeShareDoesNotExist(int shareId);
    …
 }

 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

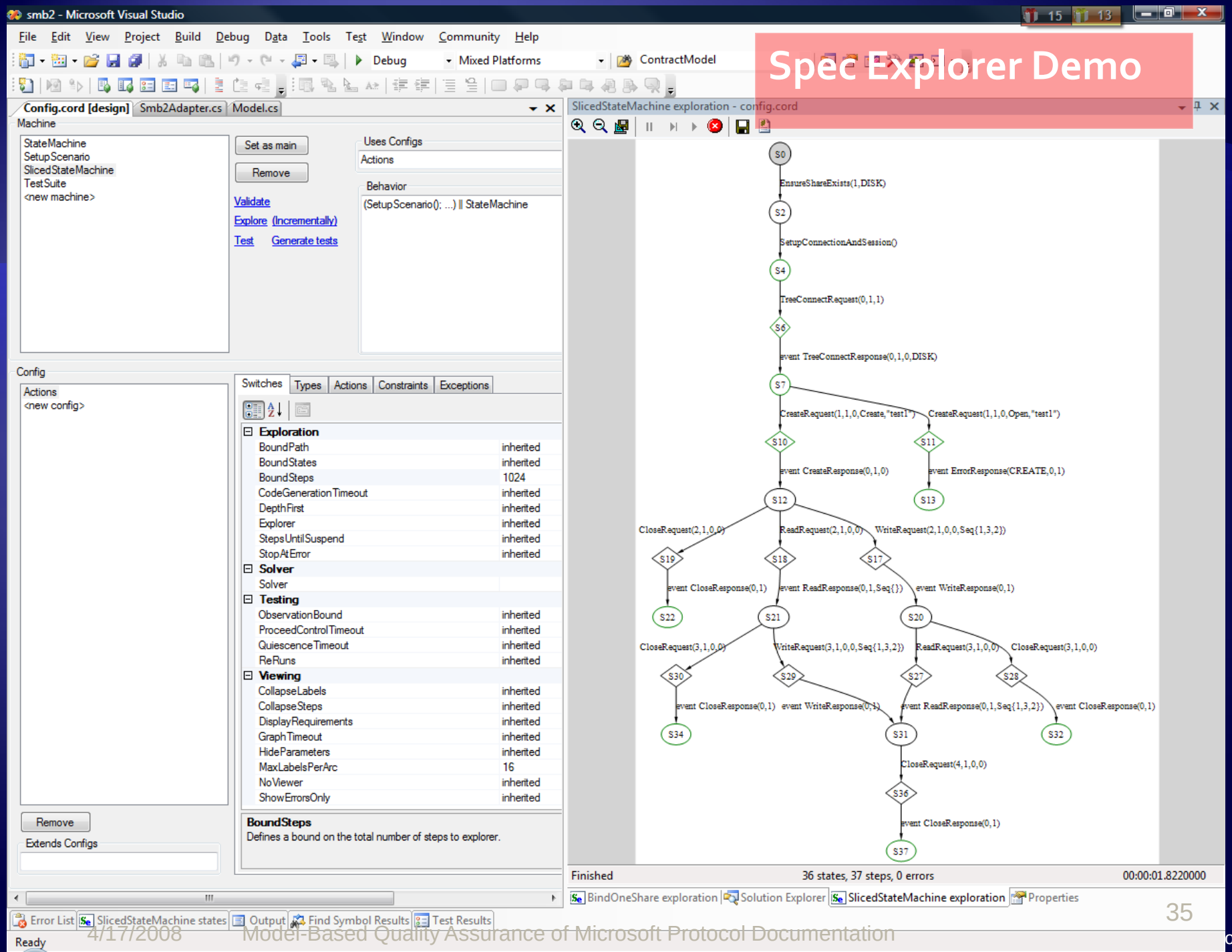
Test Selection: SMB2

```
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



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