

2014 Model-based Testing User Survey: Results



Robert V. Binder
Anne Kramer
Bruno Legeard

Contents

1	Overview	2
1.1	About the 2014 MBT User Survey	2
1.2	About this document	2
1.3	Result overview	3
2	Detailed Results	5
	Q1: When did you start using or evaluating Model-Based Testing?	5
	Q2: What do you expect from a model-based approach to testing?	6
	Q3: From your current experience, does MBT fulfil those expectations?	8
	Q4: On which test level(s) do you employ MBT?	9
	Q5: On which type(s) of testing do you employ MBT?	10
	Q6: At what stage of MBT adoption is your organization?	11
	Q7: Do you use models in other development phases, for example for requirement elicitation or system design?	12
	Q8: What type of models do you use in those other development phases?	13
	Q9: If you use models both for analysis or design and for testing activities, how different are these models?	14
	Q10: Considering all testing effort on your project,	15
	Q11: What is the subject of your model?	16
	Q12: What aspect does your MBT model focus on?	17
	Q13: How detailed is your MBT model?	18
	Q14: What notation type do you use for your MBT model?	19
	Q15: What artifacts do you generate from the model?	20
	Q16: How do you obtain the test cases from the model?	21
	Q17: What selection criteria do you apply to limit the number of generated tests?	22
	Q18: What kind of coverage is measured for your test runs?	23
	Q19: How are the tests executed?	24
	Q20: To what extent are your MBT tools integrated into the test process?	25
	Q21: To summarize, how important is the role the model plays in your MBT approach?	26
	Q22: Overall, how effective do you think MBT has been?	28
	Q23: How likely are you to continue to use MBT?	29
	Q24: I see the biggest limitation of MBT as:	30
	Q25: I see the biggest advantage of MBT as:	32
	Q26: Please provide your best estimate of the effects of MBT.	34
	Q27: On average, about how many hours of skill development were needed for a person to become a proficient MBT user?	35
	Q28: What is the general application domain of the system under test?	36
	Q29: What kind of organization sponsored / is sponsoring the Model-Based Testing project?	37
	Q30: Approximately how many persons work in this organization, in total?	38
	Q31: What was the overall process approach used for the MBT project?	39
	Q32: In the project you had in mind when filling out the previous questions, you were... ..	40

1 Overview

1.1 About the 2014 MBT User Survey

This survey is a follow up to the 2012 MBT User Survey: <http://robertvbinder.com/real-users-of-model-based-testing/> and integrates questions from a survey distributed at last year's User Conference on Advanced Automated Testing (UCAAT 2013).

It was conducted from mid-June 2014 to early August 2014. 100 model-based testing practitioners answered, providing a relevant picture of the current state of the practice in the MBT area.

The purpose of the survey was to collect data and experience on the usage of Model-based Testing. We wanted to learn how MBT users view the efficiency and effectiveness of the approach, what works, and what does not work. Some questions were more technical and aimed at validating a common MBT classification scheme.

1.2 About this document

This document summarizes the results of the 32 questions. For each question, we first provide the raw data and then a graphical representation of the result (if applicable). Beware of the fact that the number of answers varies for each question.

For numerical answers, we provided both the average and the median. The median indicates the middle of a distribution, that is, the point where there are as many answers below as above. It is less sensitive to extreme answers and usually more informative than the average.

Please, let us know if you have any comments, question or remarks regarding this MBT User Survey 2014.

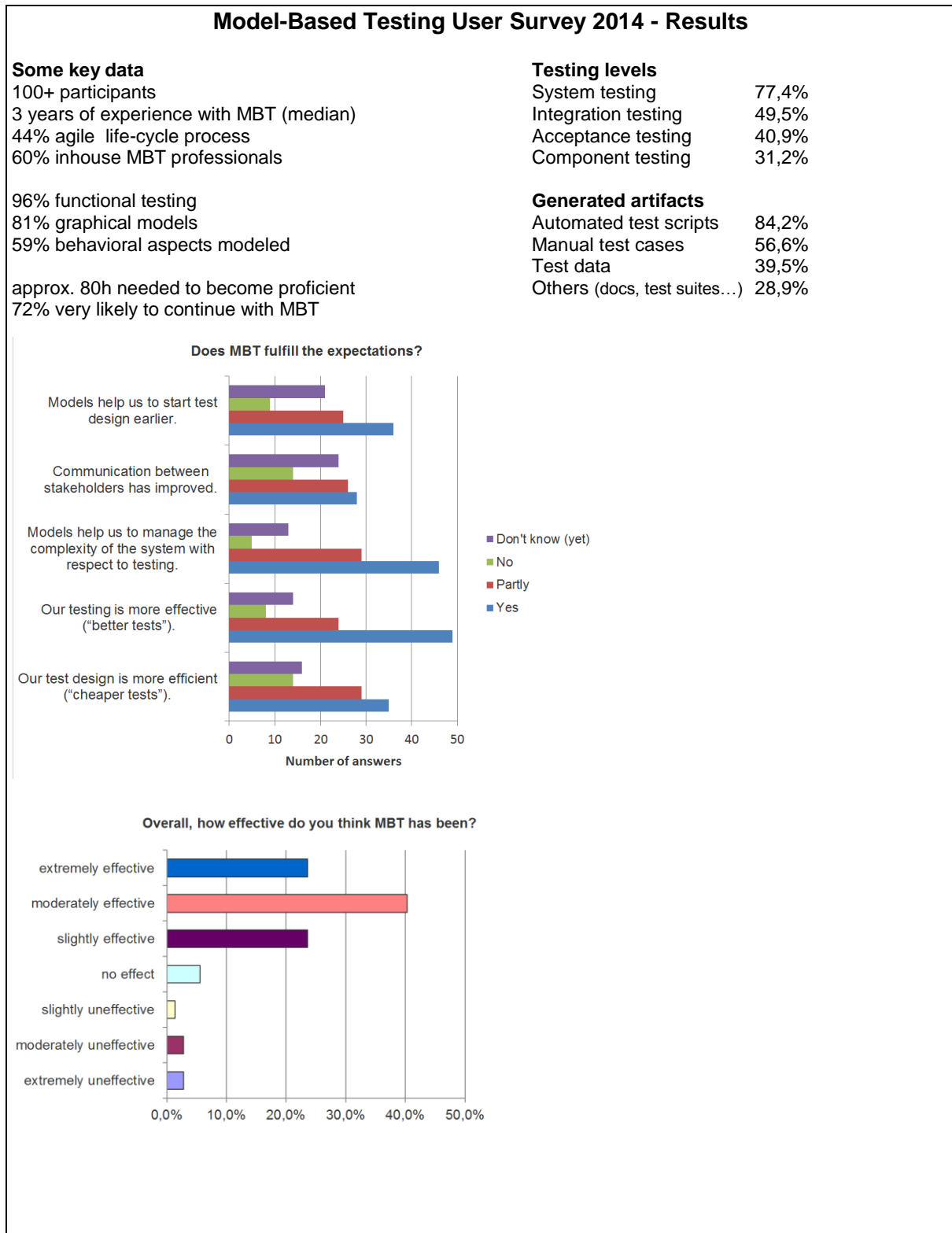
Robert V. Binder, System Verification Associates - rbinder@sysverif.com

Anne Kramer, sepp.med gmbh - anne.kramer@seppmed.de

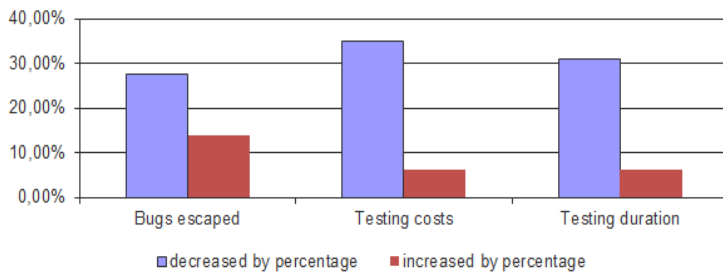
Bruno Legeard, Smartesting & University of Franche-Comté – bruno.legeard@femto-st.fr

1.3 Result overview

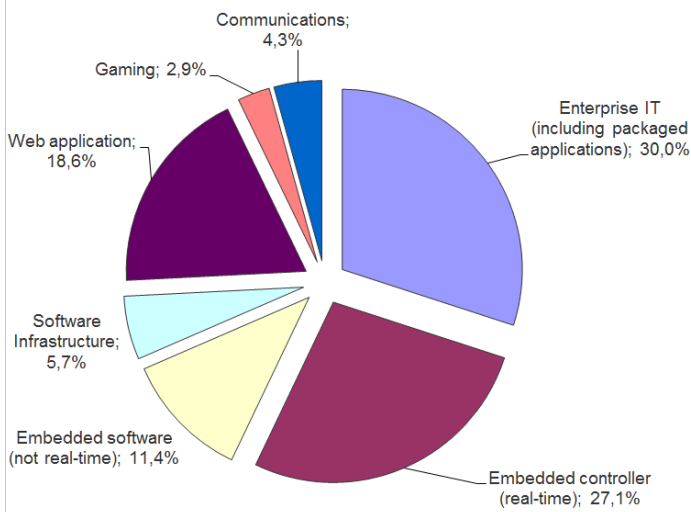
This section provides an overview on the results. Apart from the layout and two minor corrections, it corresponds to a poster we presented in October 2014 at the 2nd User Conference on Advanced Automated Testing (UCAAT) in München.



Best estimate of the effects of MBT (21 answers)



What is the general application domain of the system under test?



Biggest benefits:

test coverage, mastering complexity, automatic test case generation, reuse of models and model elements

Biggest limitations:

tool support, skills availability for MBT, resistance against change

Other interesting statements:

- MBT Models themselves used as documentation
- Tools are not user-friendly enough
- MBT is in our company globally not accepted yet.
- Only practical way to objectively assess system reliability.



SYSTEM VERIFICATION ASSOCIATES
Enabling High Assurance

Robert V. Binder
rvbinder@sysverif.com



Smartesting
femto-st
SCIENCES & TECHNOLOGIES

Bruno Legeard
bruno.legeard@smartesting.com

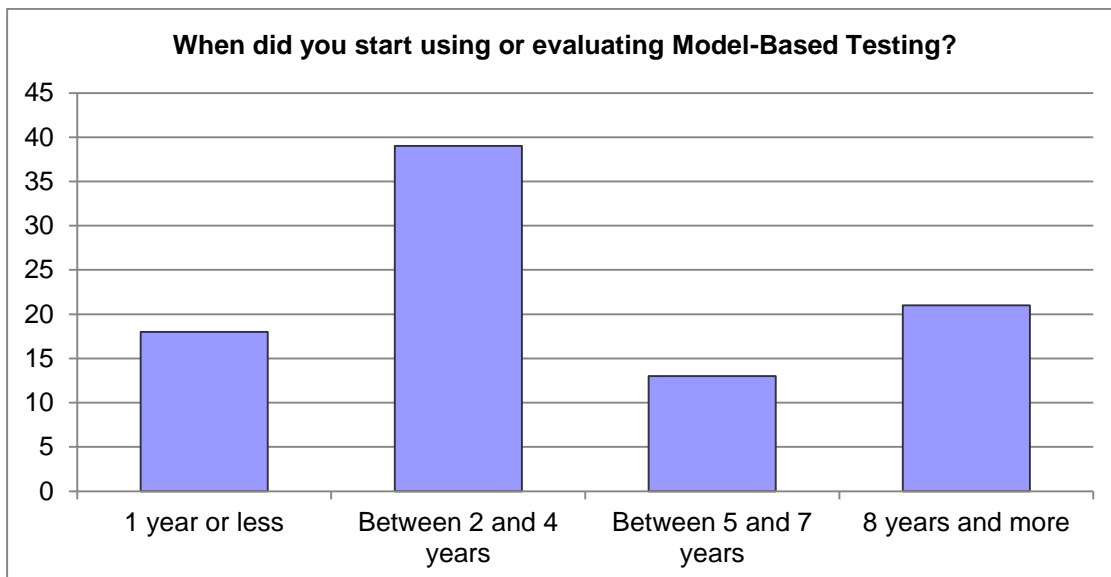


sepp.med
Qualität sichert Erfolg

Anne Kramer
anne.kramer@seppmed.de

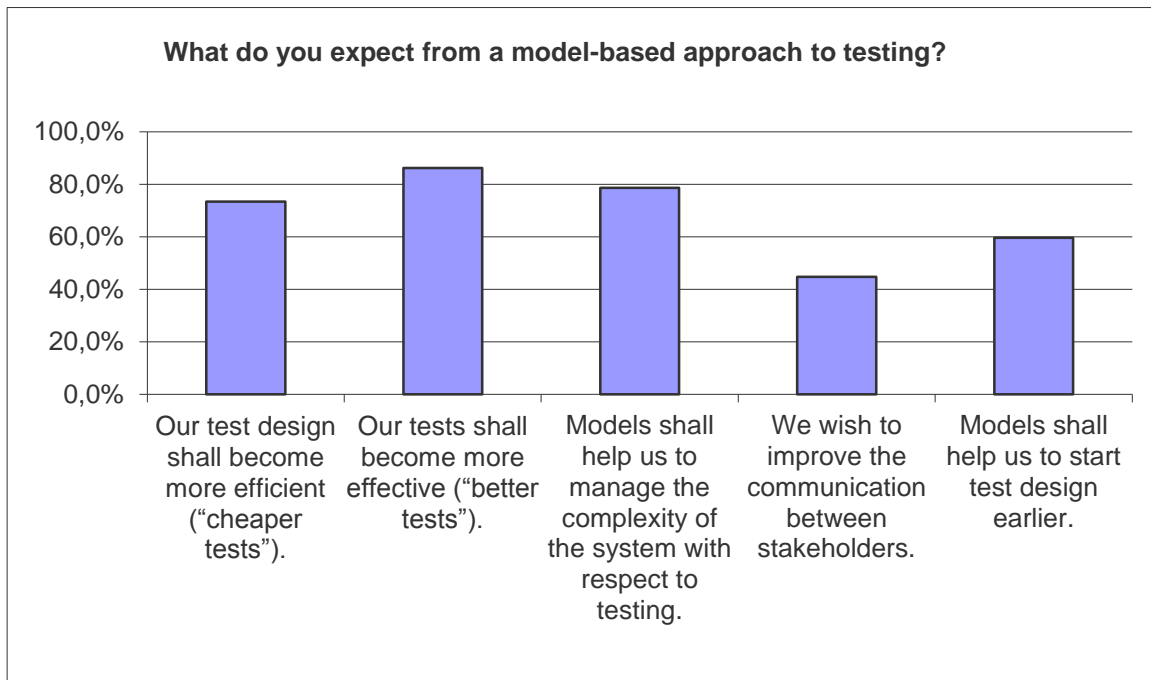
2 Detailed Results

Q1: When did you start using or evaluating Model-Based Testing?		
Answer Options	Response Average	Response Median
nn years ago	5,1	3
<i>answered question</i>		94
<i>skipped question</i>		10
1 year or less		18
Between 2 and 4 years		39
Between 5 and 7 years		13
8 years and more		21
Minimum value		0
Maximum value		34



Q2: What do you expect from a model-based approach to testing?

Answer Options	Response Percent	Response Count
Our test design shall become more efficient ("cheaper tests").	73,4%	69
Our tests shall become more effective ("better tests").	86,2%	81
Models shall help us to manage the complexity of the system with respect to testing.	78,7%	74
We wish to improve the communication between stakeholders.	44,7%	42
Models shall help us to start test design earlier.	59,6%	56
Other expectations:		25
answered question		94
skipped question		10



Other expectations:

- Our scripted tests do not change for many years. Test Models shall provide more flexibility in the creation of test scripts by using different test case generation strategies according to the current needs (different robustness test, tests for changed or critical requirements, etc.)
- Models shall help us improve reporting on testing activities.
- Executable model paradigm should enable us to create self-testing mechanisms.
- I expect that there must be some common way to integrate MBT with QTP & UFT
- It should be possible to use one generic model in different tools.
- we would like to integrate matelo with QTP
- An inspiring challenge for the testers
- MBT helps us to improve change request process
- Reuse of test artefacts throughout the different test level
- Involvement of additional stakeholders in test design activities and in evaluation of the quality of test design"
- Being able to run more test runs on a broader range of devices to cover our most basic test cases. This will give us a hint if the build(s) are worth QA:ing. In the long run, proper TA tests that cover many basic test cases could be the way towards CI/CD.

- MBT supports test automation
- And perhaps some things more you didn't mention ;-)
- MBT is capable to also validate the test basis
- Massive increase of reuse
- Enable us to work in a true team (compositional)
- add variations in automated regression test
- Reducing Time for test script preparation Easier Test scope selection, test coverage
- -> ROI Report -> Test Coverage Report -> Models Portability
- Serves to automate the test design.
- All testing uses models. Number-based algebra is just algebra; just like model-based testing is just that: testing.
- improving maintainability of tests whilst watching the s/w design changing vs its implementation...
- Models and automatic test generation shall reduce the amount of human errors.
- Improved coverage of requirement readability of coverage
- Increased productivity Improved product reliability (new type of bugs, Increased test coverage) Better test script maintenance Agility (Easily react to new feature changes, Reusability of test semantics, Early test engagement, Drive quality upstream) Increased employee satisfaction (challenging, new horizon, fun)
- Improve time to market especially in agile projects

Q3: From your current experience, does MBT fulfil those expectations?

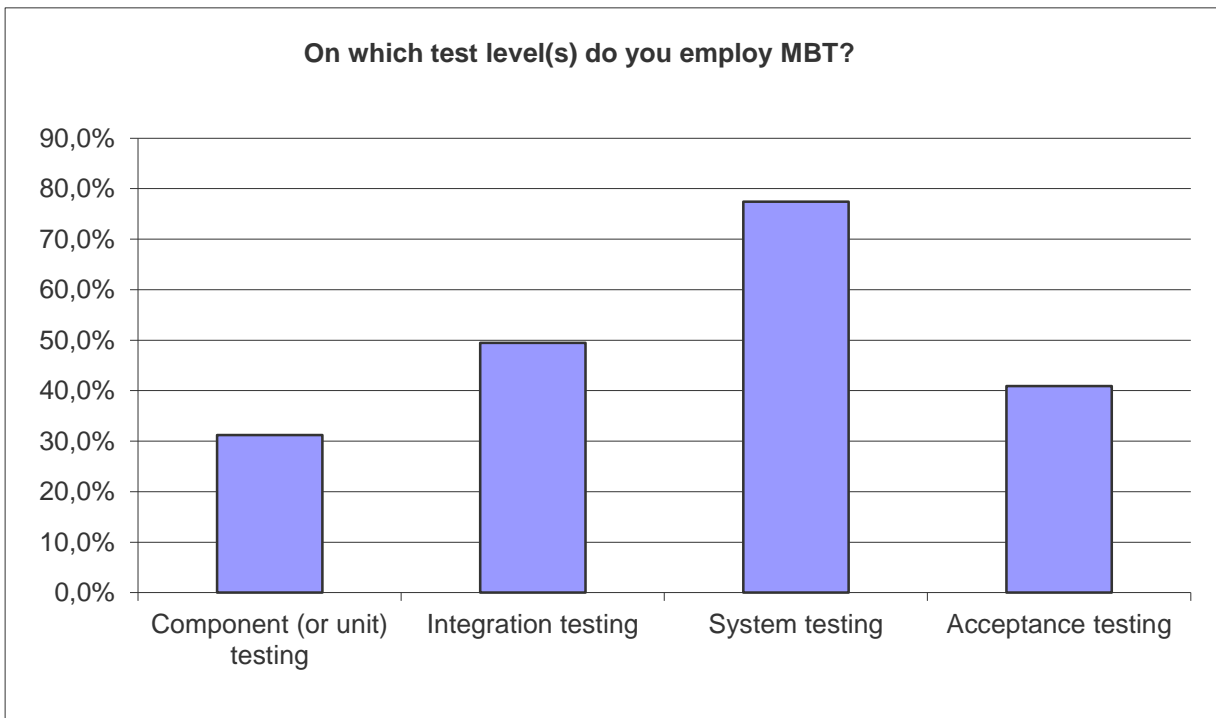
Answer Options	Yes	Partly	No	Don't know (yet)	Rating Average	Response Count
Our test design is more efficient ("cheaper tests").	35	29	14	16	1,73	94
Our testing is more effective ("better tests").	49	24	8	14	1,49	95
Models help us to manage the complexity of the system with respect to testing.	46	29	5	13	1,49	93
Communication between stakeholders has improved.	28	26	14	24	1,79	92
Models help us to start test design earlier.	36	25	9	21	1,61	91
If MBT does not satisfy the expectations you placed in it, please indicate the reason						20
answered question						95
skipped question						9

Other reasons:

- Test Models do not solve our challenge to improve the quality of system / software requirements.
- the effort is very high
- Using SpecExplorer, writing the model too early with little insight into the implementation details, can often lead to significant rework on the model. Additionally, while very beneficial for complex combinatorial challenges, we often find it to be overkill for simpler aspects of the application under test. It needs to be judiciously applied.
- Stakeholders are focused on their own box of objectives. It takes a certain type of inquisitive personality to see other views... Technology is not as efficient in opening human minds.
- No great tooling option exists. SpecExplorer comes the closest to what we need, but it's way too complicated.
- can we use the same model for generating tests and generating production code?
- Lack of experience
- I don't see the advantages of it !
- we would like to integrate matelo with QTP, but we don't know how to integrate
- Devs should learn MBT
- Really difficult to implement MBT if devs as general are not working with models
- Often prerequisites to successfully use models aren't met, so MBT doesn't fully live up to expectations. But early communication with 'others' always works through models.
- The efficiency of MBT approaches is usually reached in an iterative test process; that's why it cannot be said per se that MBT leads to better efficiency. Improvement of communication highly depends on the representation of the model;
- Difficulty in communication: MBT is in our company globally not accepted yet.
- -> While optimising missing some important Test scenarios -> Model Portability is not available between tools -> Test Coverage Report -> Testing Effort Savings Report -> MBT ROI Report
- The modeling aspect though we have done a lot to shield the complexities, is too complex for most of the stakeholders. The dont gave a clue. Even most of the testers. Conventional performance testing is much easier for them to oversee.
- Have tried hard but models are too hard to create/maintain at enough granularity to create valuable tests. Also suffer badly from being downstream from development, even when being member of the dev team because when functionality iterates too big changes are required on the models.
- Model-based testing is just testing. It doesn't "improve" my testing at all, because I've always used models - just like everyone in testing.
- incorrect models that do not match intentions and specifications is the biggest threat to MBT. In addition, architectural erosion makes many models useless.
- Not yet fully acceptatble by some testers (is still not mainstream)

Q4: On which test level(s) do you employ MBT?

Answer Options	Response Percent	Response Count
Component (or unit) testing	31,2%	29
Integration testing	49,5%	46
System testing	77,4%	72
Acceptance testing	40,9%	38
Other (please specify)		3
answered question		93
skipped question		11

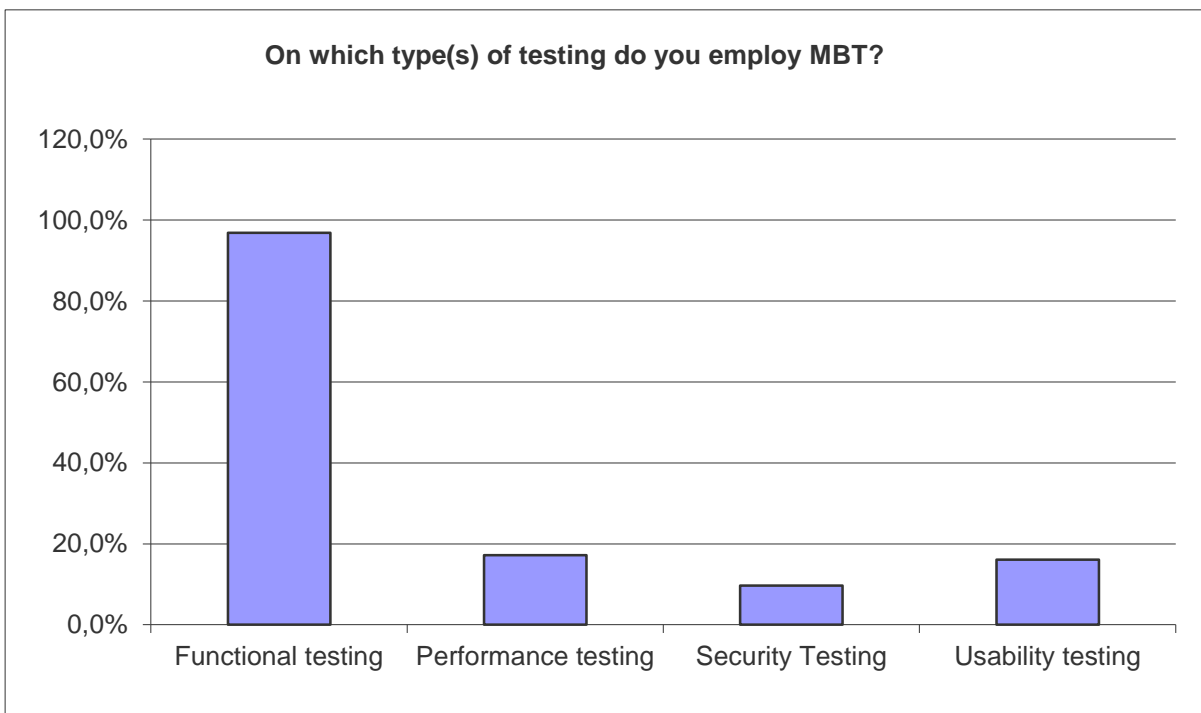


Others:

- We aim to employ MBT on integration and system test levels
- Processus testing
- There are no test levels. The above are more types of testing rather than levels. Using "Levels" as terminology seems to imply that one follows the other, or one is on top of the other. They don't and they aren't. And I use models in all my testing.

Q5: On which type(s) of testing do you employ MBT?

Answer Options	Response Percent	Response Count
Functional testing	96,8%	90
Performance testing	17,2%	16
Security Testing	9,7%	9
Usability testing	16,1%	15
Others		8
answered question		93
skipped question		11

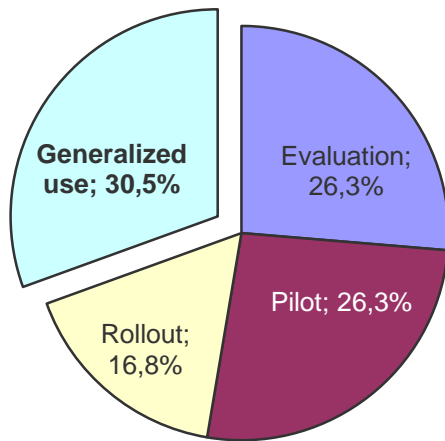


Others:

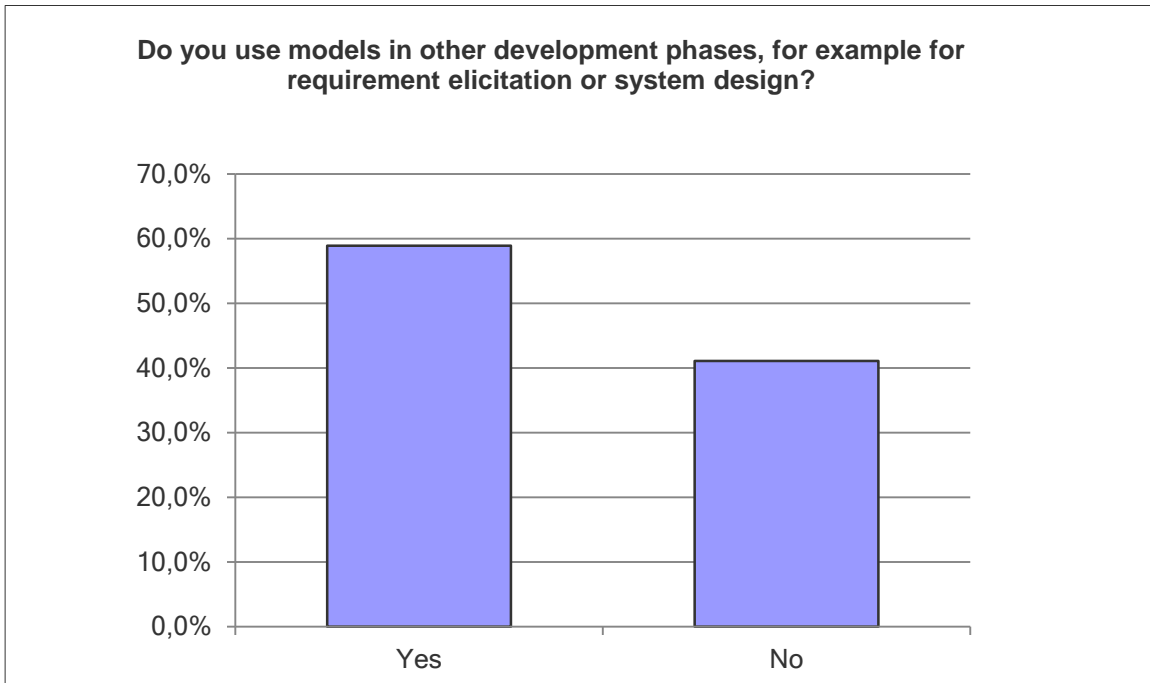
- We aim to employ MBT on functional type of testing.
- non functional requirements
- Robustness testing, and some other testing of non-functional requirements
- Unitary testing
- early validation
- More specifically: regression testing
- These "types" of testing, being the kind of tests to find information of the quality characteristic/attribute its name contains, are only a fragment. I use models for all my testing.
- Safety

Q6: At what stage of MBT adoption is your organization?		
Answer Options	Response Percent	Response Count
Evaluation	26,3%	25
Pilot	26,3%	25
Rollout	16,8%	16
Generalized use	30,5%	29
answered question		95
skipped question		9

At what stage of MBT adoption is your organization?

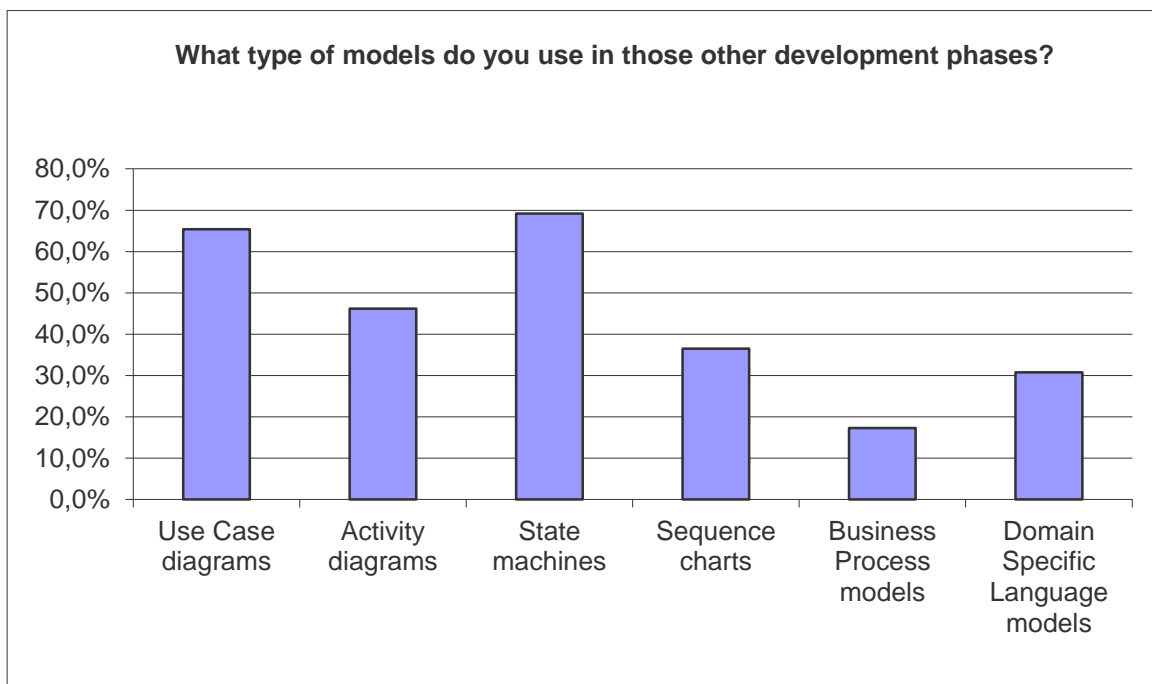


Q7: Do you use models in other development phases, for example for requirement elicitation or system design?		
Answer Options	Response Percent	Response Count
Yes	58,9%	56
No	41,1%	39
answered question		95
skipped question		9



Q8: What type of models do you use in those other development phases?

Answer Options	Response Percent	Response Count
Use Case diagrams	65,4%	34
Activity diagrams	46,2%	24
State machines	69,2%	36
Sequence charts	36,5%	19
Business Process models	17,3%	9
Domain Specific Language models	30,8%	16
Others (please specify):		10
answered question		52
skipped question		52

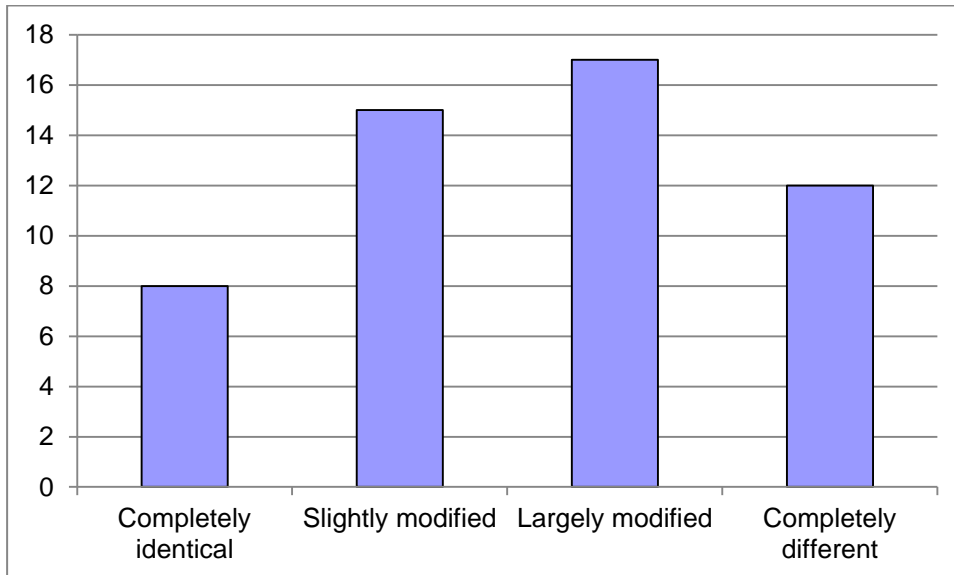


Others:

- Formal models such as Petri Nets
- Custom models in tool like matlab
- At least every kind of UML diagram type.
- Simulink
- Class models
- Rule based models
- pseudo code, decision tables. Essentially: anyt artifact that is anambiguous and can be processed by a computer
- Behavior Trees following the Behavior Modelling Language
- Analytic queuing models (the mBrace Model)
- Mind maps Similar programs Requirements documents all our oracles are models

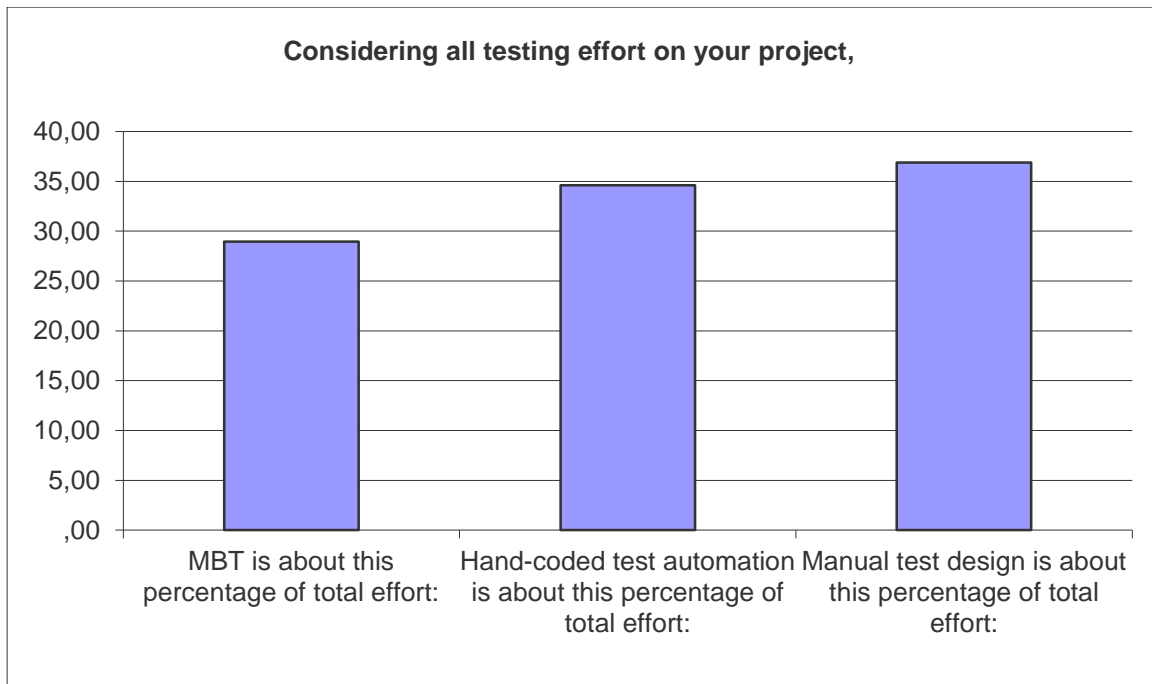
Q9: If you use models both for analysis or design and for testing activities, how different are these models?

Answer Options	Completely identical	Slightly modified	Largely modified	Completely different
Degree of redundancy	8	15	17	12
<i>answered question</i>				52
<i>skipped question</i>				52



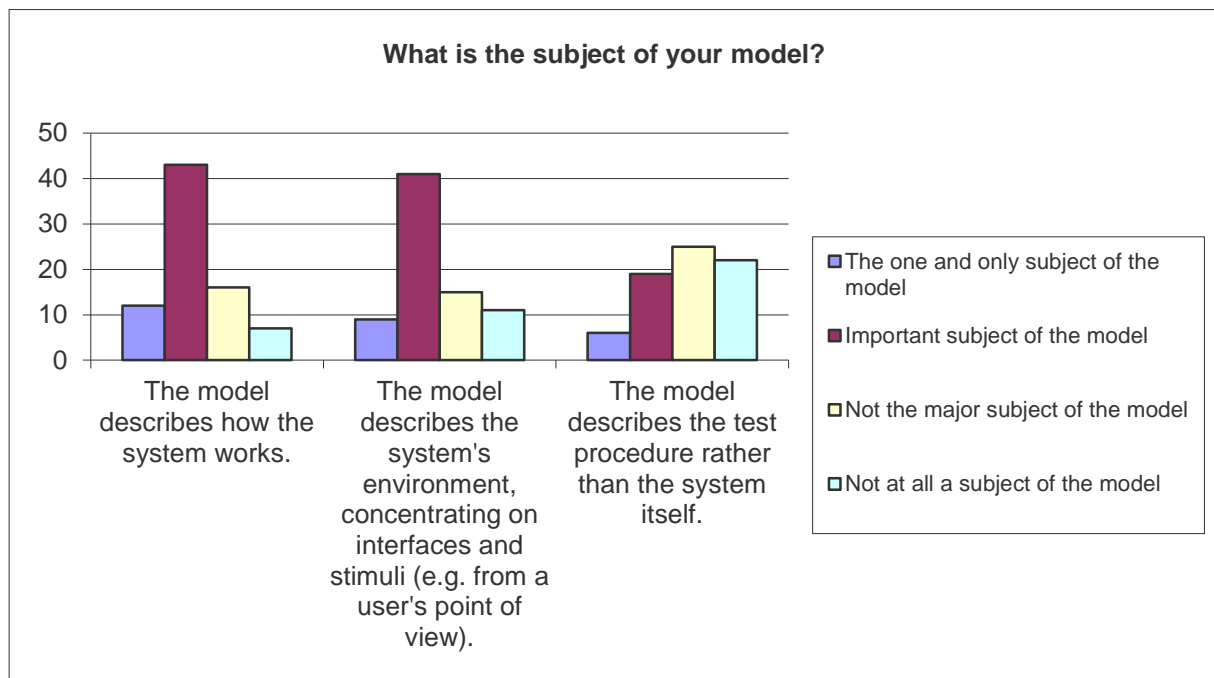
Q10: Considering all testing effort on your project,...

Answer Options	Response Average	Response Total	Response Count
MBT is about this percentage of total effort:	28,95	2.200	76
Hand-coded test automation is about this percentage of total effort:	34,62	2.562	74
Manual test design is about this percentage of total effort:	36,89	2.693	73
answered question			76
skipped question			28



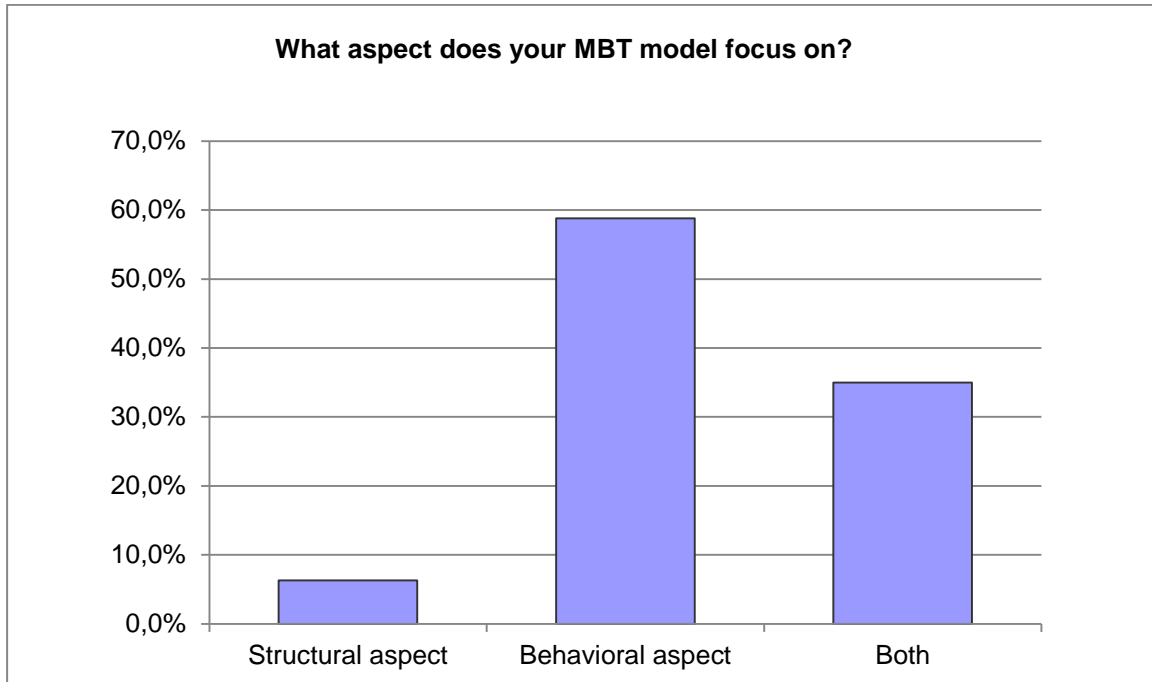
Q11: What is the subject of your model?

Answer Options	Not at all a subject of the model	Not the major subject of the model	Important subject of the model	The one and only subject of the model	Response Count
The model describes how the system works.	7	16	43	12	78
The model describes the system's environment, concentrating on interfaces and stimuli (e.g. from a user's point of view).	11	15	41	9	76
The model describes the test procedure rather than the system itself.	22	25	19	6	72
answered question					80
skipped question					24



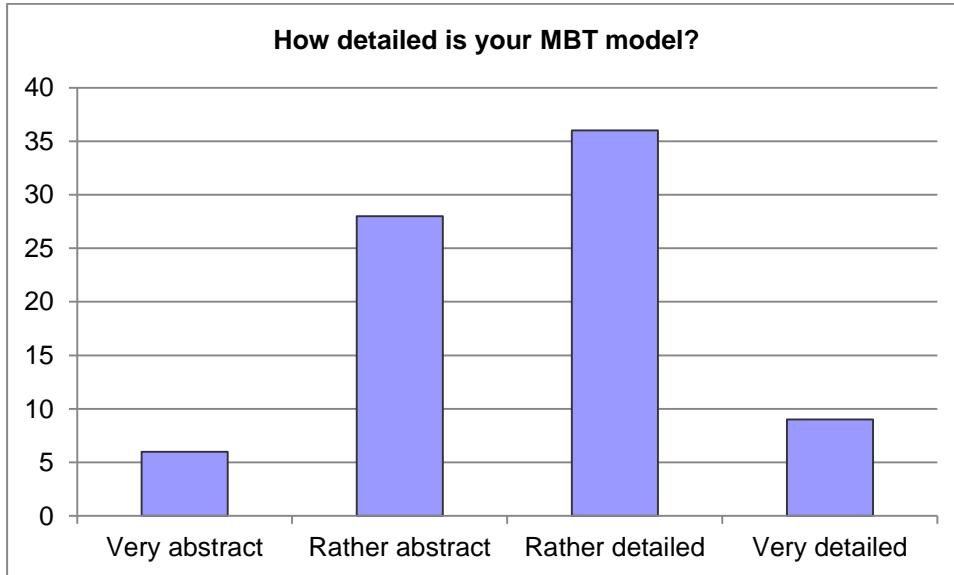
Q12: What aspect does your MBT model focus on?

Answer Options	Response Percent	Response Count
Structural aspect	6,3%	5
Behavioral aspect	58,8%	47
Both	35,0%	28
answered question		80
skipped question		24



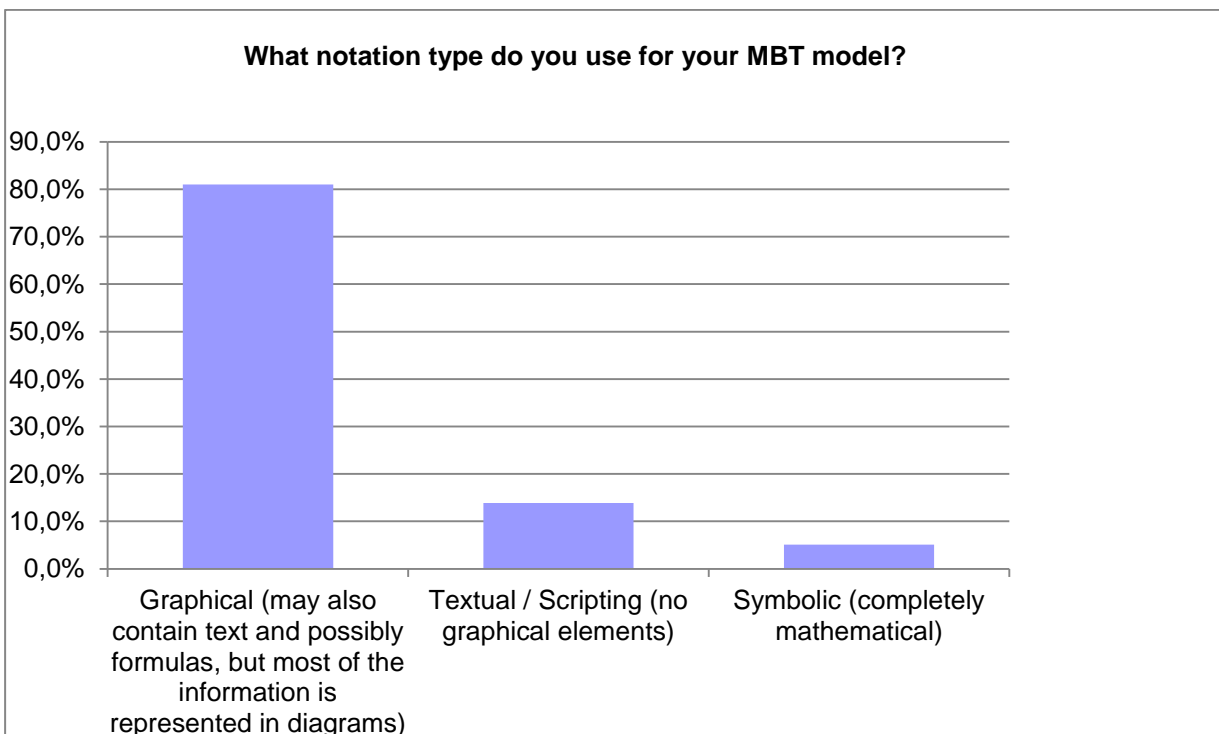
Q13: How detailed is your MBT model?

Answer Options	Very abstract	Rather abstract	Rather detailed	Very detailed
Degree of abstraction	6	28	36	9
<i>answered question</i>				79
<i>skipped question</i>				25



Q14: What notation type do you use for your MBT model?

Answer Options	Response Percent	Response Count
Graphical (may also contain text and possibly formulas, but most of the information is represented in diagrams)	81,0%	64
Textual / Scripting (no graphical elements)	13,9%	11
Symbolic (completely mathematical)	5,1%	4
Other (please specify)		12
answered question		79
skipped question		25

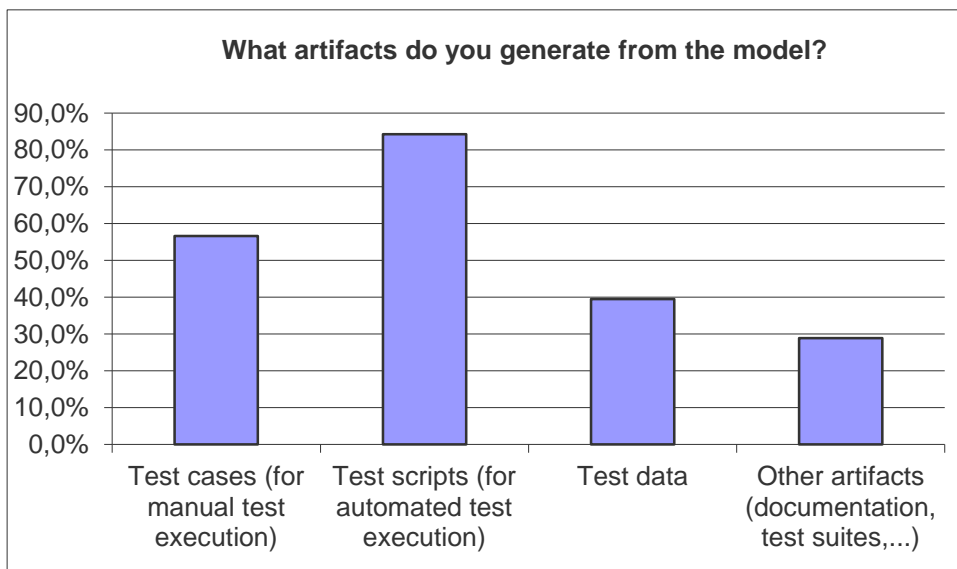


Others:

- SpecExplorer
- most of the information textual, with some diagrams
- SpecExplorer/C#
- Domain-Specific Language spreadsheet
- We use more than 1 model per system under the test
- Not scripting, our MBT tool imports XML representations of graphical and/or textual models.
- We heavily combine diagrams with usual UI widgets like tables, trees, buttons, text fields etc. to represent and modify the model
- UTP
- ... Some models are graphical, others are textual, some symbolic, some are never more than mental models, some are fully automated oracles that model a specific functionality, ...
- all three!
- Mainly symbolic (UML-OCL), but also graphical (UML class/state diagram, BPMN) and also textual for adaptation layer specification
- textual & Graphical

Q15: What artifacts do you generate from the model?

Answer Options	Response Percent	Response Count
Test cases (for manual test execution)	56,6%	43
Test scripts (for automated test execution)	84,2%	64
Test data	39,5%	30
Other artifacts (documentation, test suites,...)	28,9%	22
If "Other", please indicate the generated artifacts:		17
answered question		76
skipped question		28



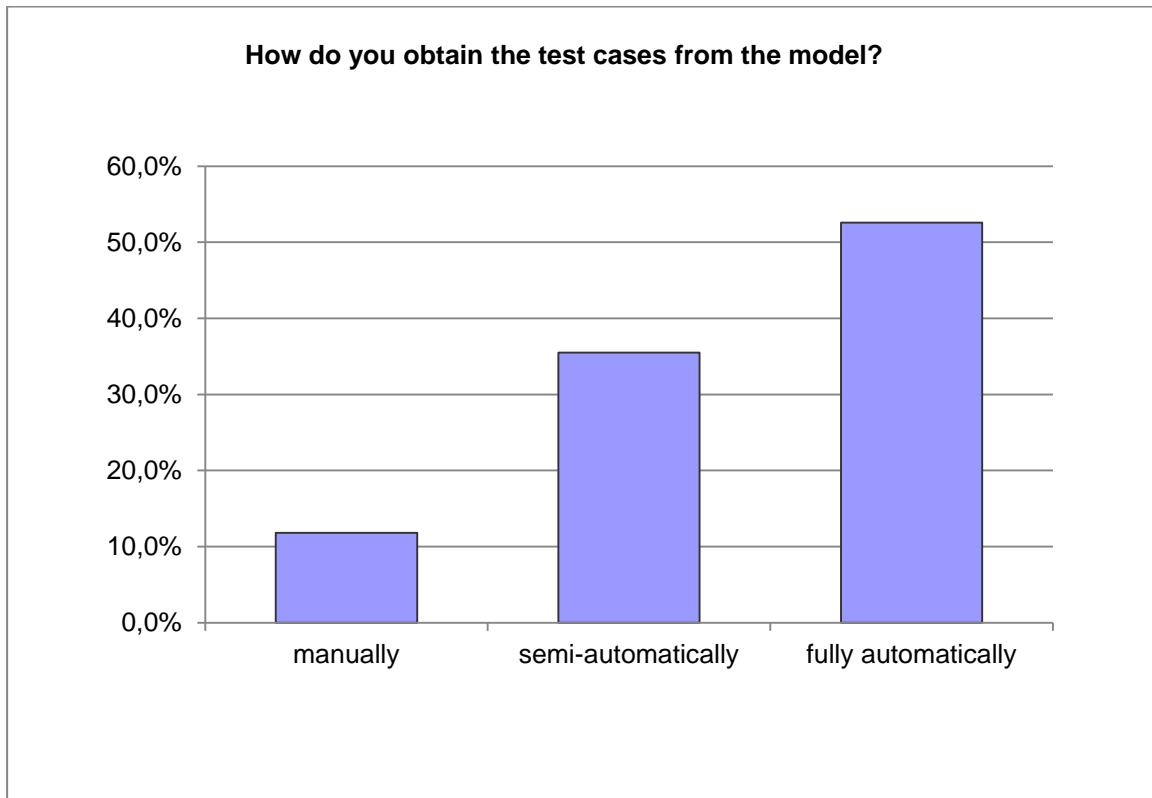
Other artifacts:

- Success criteria
- not anymore
- model and data are enterd in the model Model added data, create only the test cases and submit them This tool is perfect !!! for testing adminitrive processes!!
- Models themselves used as documentation
- test plan, test specification
- documentation of verified and observed properties
- reports, metrics, adaptation layer stubs
- Test documentation, Requirements coverage
- Visualization of test case and test coverage. Defect location.
- Test suites, reports
- Test documentation
- user story/requirements coverage
- I don't understand this question. Whether or not there are artifacts has no relevance as to whether these artifacts are "generated from" a model.
- test suites test harness coverage metrics test run reports
- Docs
- whole test suites
- Test Plan (SUT, configuration)

Q16: How do you obtain the test cases from the model?

Answer Options	Response Percent	Response Count
manually	11,8%	9
semi-automatically	35,5%	27
fully automatically	52,6%	40
answered question		76
skipped question		28

This question aimed at the test artifact generation process. For tool integration, see question 20.



Q17: What selection criteria do you apply to limit the number of generated tests?

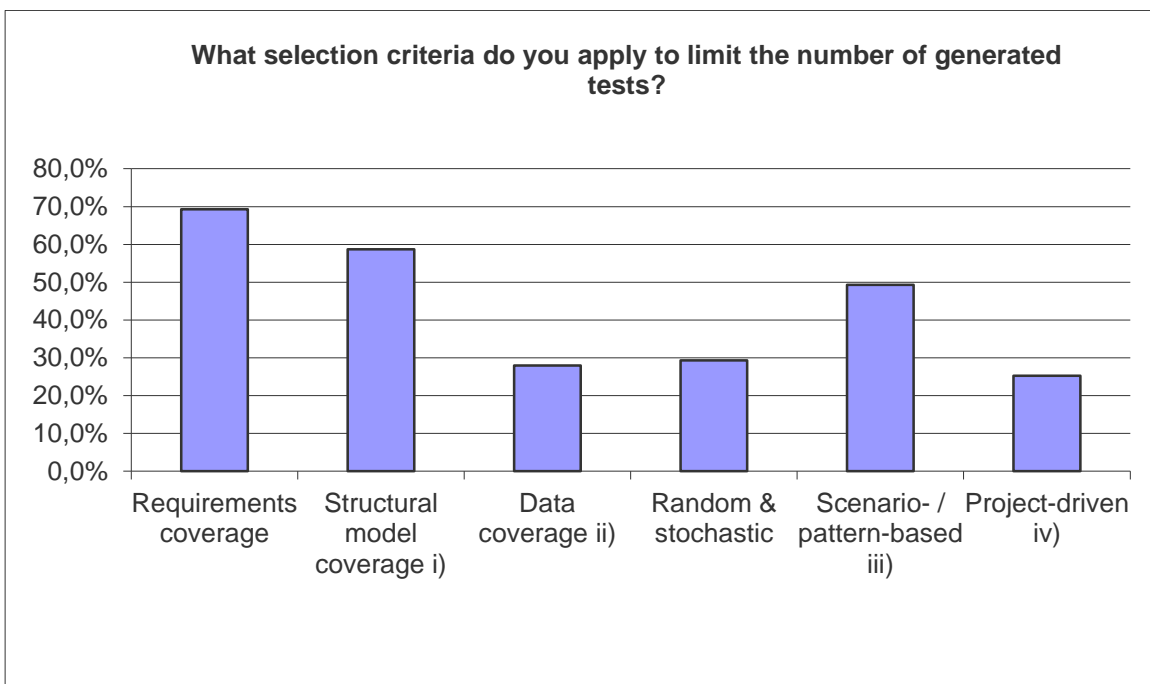
Answer Options	Response Percent	Response Count
Requirements coverage	69,3%	52
Structural model coverage i)	58,7%	44
Data coverage ii)	28,0%	21
Random & stochastic	29,3%	22
Scenario- / pattern-based iii)	49,3%	37
Project-driven iv)	25,3%	19
Others:		6
answered question		75
skipped question		29

i) Structural model coverage is a generic term for a variety of coverage criteria that are based on the internal structure of the model (e.g. transition coverage).

ii) Data coverage focuses on the equivalence partitions of input and output data, possibly combined with a boundary value analysis.

iii) Scenarios or patterns are specific paths through the model that correspond to frequently used functionality or fault-based scenarios.

iv) Project-driven test case selection criteria uses specific additional information added to the model (e.g. risk, priorities...) to drive test generation to achieve specific test objectives for the project.

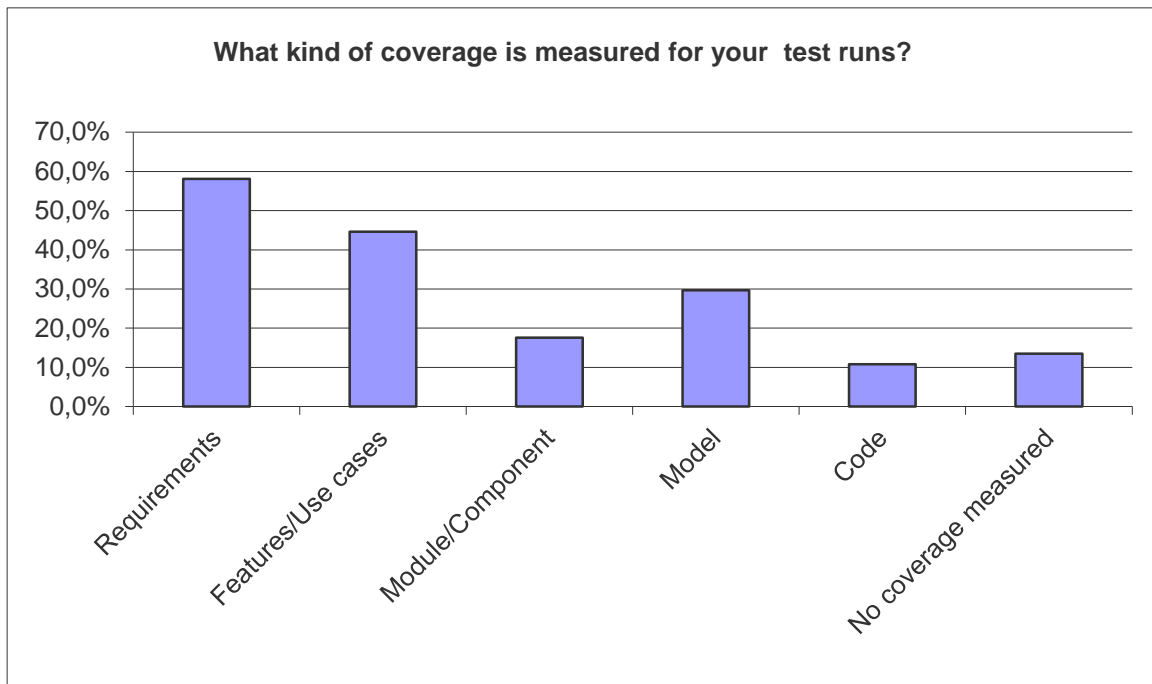


Others:

- criteria specific to the model being used
- none
- Test techniques based and depending the model. Fi. MCDG, MCC, DC, pairwise
- test requirements coverage (which is similar to requirements coverage)
- Depends on the context
- Test Objective Charter

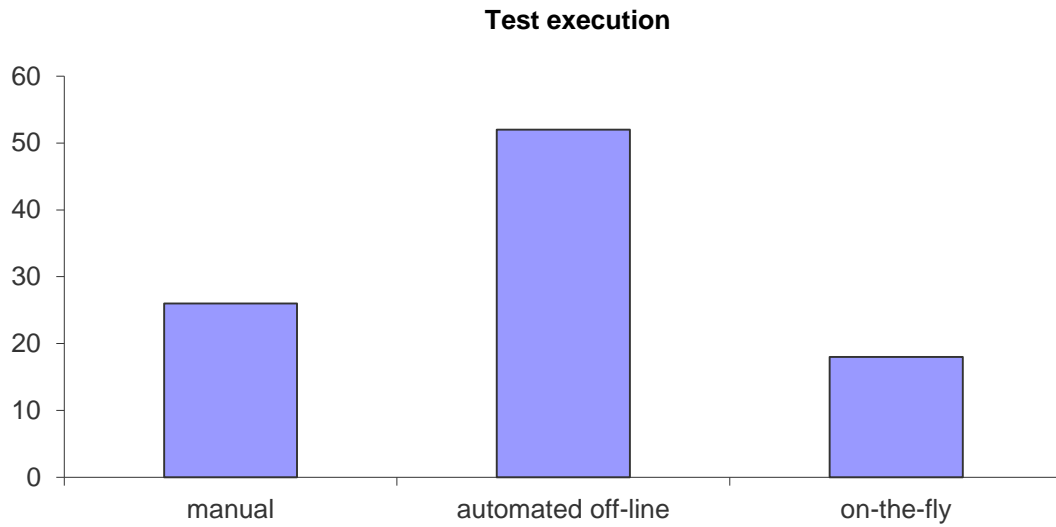
Q18: What kind of coverage is measured for your test runs?

Answer Options	Response Percent	Response Count
Requirements	58,1%	43
Features/Use cases	44,6%	33
Module/Component	17,6%	13
Model	29,7%	22
Code	10,8%	8
No coverage measured	13,5%	10
answered question		74
skipped question		30



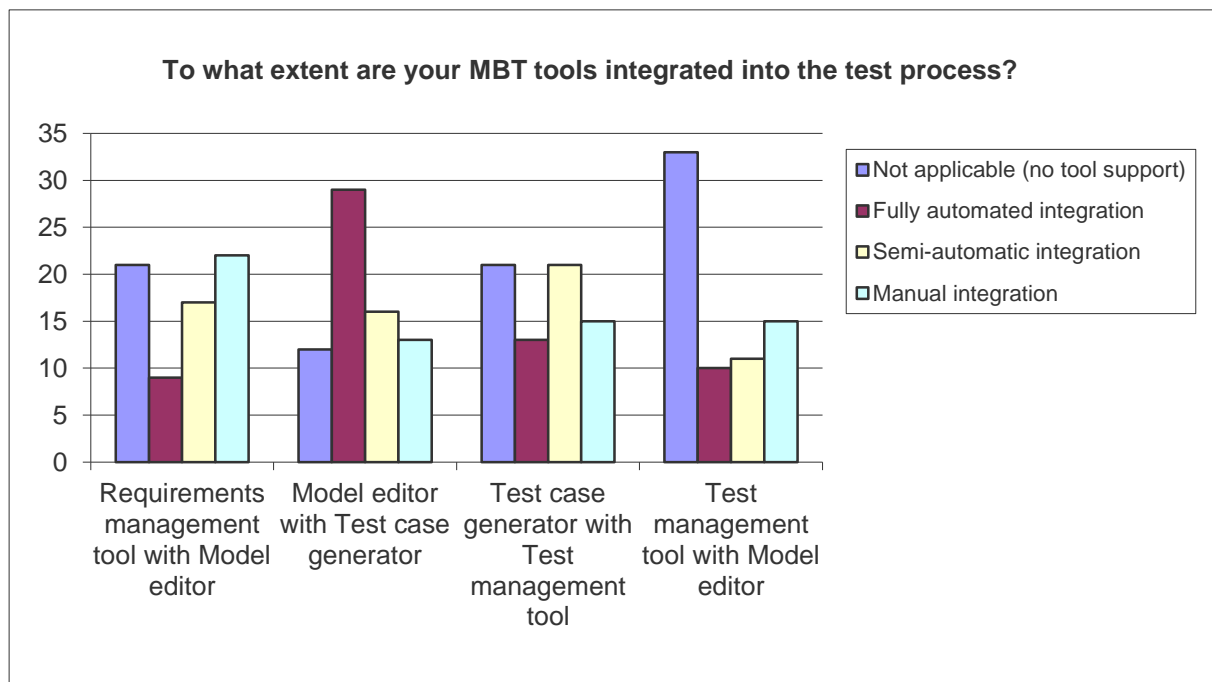
Q19: How are the tests executed?

Answer Options	Response Percent	Response Count
manually	34,7%	26
automatically off-line (first generated, then executed)	69,3%	52
automatically on-the-fly (generated and executed in one step)	24,0%	18
answered question		75
skipped question		29



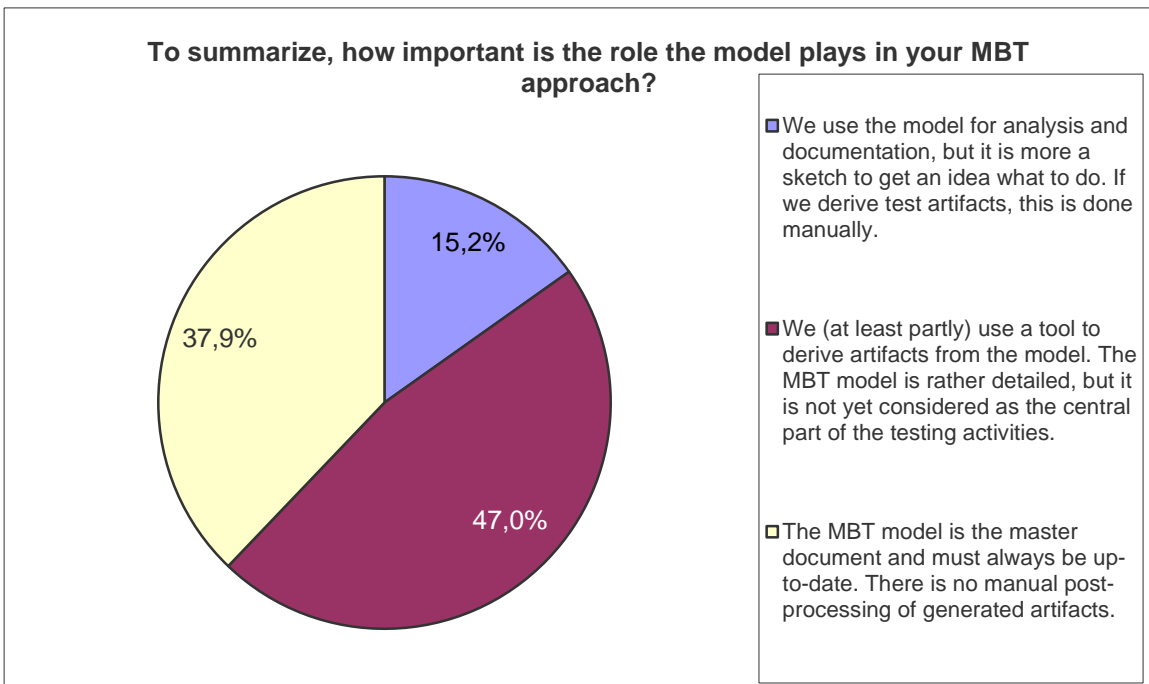
Q20: To what extent are your MBT tools integrated into the test process?

Answer Options	Manual integration	Semi-automatic integration	Fully automated integration	Not applicable (no tool support)	Response Count
Requirements management tool with Model editor	22	17	9	21	69
Model editor with Test case generator	13	16	29	12	70
Test case generator with Test management tool	15	21	13	21	70
Test management tool with Model editor	15	11	10	33	69
answered question					71
skipped question					33



Q21: To summarize, how important is the role the model plays in your MBT approach?

Answer Options	Response Percent	Response Count
We use the model for analysis and documentation, but it is more a sketch to get an idea what to do. If we derive test artifacts, this is done manually.	15,2%	10
We (at least partly) use a tool to derive artifacts from the model. The MBT model is rather detailed, but it is not yet considered as the central part of the testing activities.	47,0%	31
The MBT model is the master document and must always be up-to-date. There is no manual post-processing of generated artifacts.	37,9%	25
If you are not happy with any of the three answers, please tell us in your own words:		10
answered question		66
skipped question		38



Other answers:

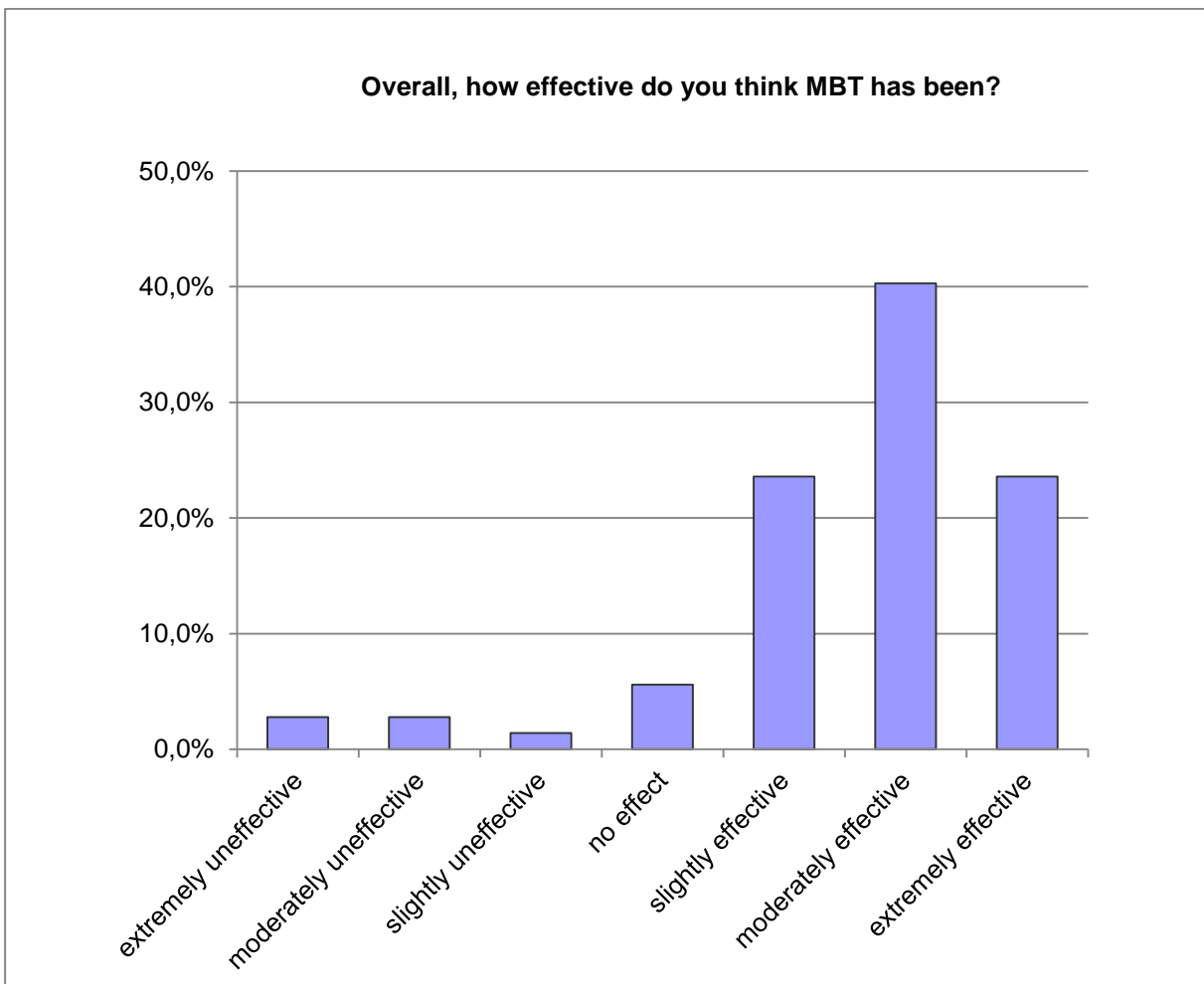
- Some extra test still done by hand. And we use it now for one complex administrative process
- We only generate models when faced with complex combinatorial test challenges. In that case, the model is the master document and all tests are generated directly from the model. Any changes require an update to the model.
- There are some manual post-processing of generated artifacts depending on the situation.
- Still evaluating approach
- The 3rd answer is correct in general, however, there is still some manual processing required, mainly in the adaptation stubs, since the test model does usually not cover all the technical details to generate the adaptation layer completely. However, test cases and test artifacts should and are not touched any more once they are generated
- MBT is an interesting approach used for testing certain features. It is not rolled out across the org. though. It is highly beneficial for regression test purposes. It eases manual efforts

and helps in preventing human errors. MBT based test design in earlier phases is particularly helpful to assure a smooth testing process.

- Why are all questions related to "the" model? If you only use one model, no matter how "good/complete/correct", your testing will be limited to testing *that model* instead of the software you're testing.
- model is master but post processing is possible
- We are delivering Test Plans and Test suites from the customers' specifications. We are not involved into development process, but only in test process. The role MBT model is essential in our methodology and delivery process
- The MBT model is rather detailed and serves as complete test design, but there is no dedicated integration of a generation algorithm. Partly, test execution is done manually with the MBT model

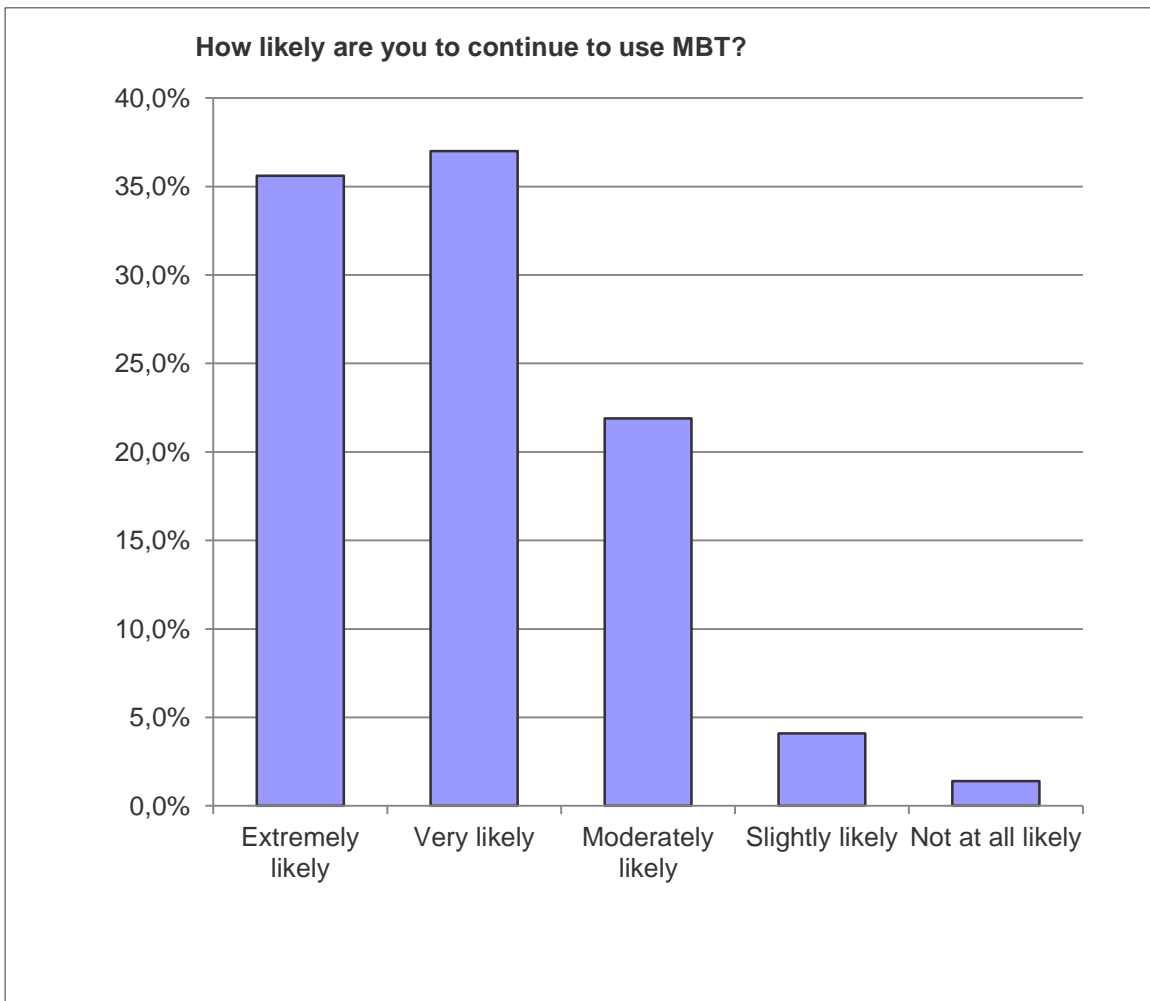
Q22: Overall, how effective do you think MBT has been?

Answer Options	Response Percent	Response Count
extremely ineffective	2,8%	2
moderately ineffective	2,8%	2
slightly ineffective	1,4%	1
no effect	5,6%	4
slightly effective	23,6%	17
moderately effective	40,3%	29
extremely effective	23,6%	17
answered question		72
skipped question		32



Q23: How likely are you to continue to use MBT?

Answer Options	Response Percent	Response Count
Extremely likely	35,6%	26
Very likely	37,0%	27
Moderately likely	21,9%	16
Slightly likely	4,1%	3
Not at all likely	1,4%	1
answered question		73
skipped question		31



Q24: I see the biggest limitation of MBT as:

Answer Options	Response Count
	51
answered question	51
skipped question	53

Answers:

- gap between abstraction level of model and code which requires translating test cases into executable scripts.
- Tooling support - tools are not user-friendly enough; tool prices are high
- complex data entry to specify model
- Supporting tools and tool to tool data interchange
- Implementation within legacy projects (thousands of lines of scripts available - too much to transfer to test models)
- adoption and understanding
- collaboration
- complexity to get in toolchain, and Interfaces to other Software. Research and innovations are to Abstract and for companies mostly not applicable
- Certain skills required to create and maintain model, absence of tools that cover enough viewpoints, effort required to build a useful model for legacy applications
- None, anymore. There is a MBT tool that can be used in the administrative environment
- Skills availability for MBT
- Applying it to incorrectly when authoring a model is overkill. For simplistic functionality, authoring a model actually costs more than writing static automated tests.
- steep learning curve computationally expensive test generation algorithms
- The problem with test case explosion and the maintenance of models.
- Lack of detailed execution semantics definition for models in general. Lack of proper definition of physics, control theory, data, and network at once.
- get users to deal with tool support
- it only generate the scripts that we include in setting functions, and those generated scripts can not be directly executed in QTP or UFT without there appropriate object repository
- Different in Modelling methods tools, non portability
- Different model for different objectives. Hi amount to generate the model. Who proofs that the model fullfills all requirements?
- The agile way of working
- Understanding and acceptance of the MBT concept throughout the whole organization
- Change Management
- acceptance as a method
- acceptance due to likely efforts
- heavy to implement. Create inter-model use case.
- Not user friendly enough
- cumbersome to keep coplex models/tests updated
- debugging
- Organizational challenges when rest of organization is not "thinking in models"
- none, but current MBT is limited regarding V&V due to separation of domains, especially separation of functional/behavioural and performance aspects
- On-boarding by clients who do not fully understand the approach and benefits
- An easy way to model and align the business risk with areas of deep testing
- Joint cooperating in the software life cycle
- Organizational/human resistence to change
- the lack of education of test analysts (or test engineers)
- Scalability in terms of complexity
- Reason in embedded systems not only about algebraic equations (i.e. look up tables) but also on plant behaviour (differential equations)

- a model is always a limited, partial description of a system
- Time consuming to utilize; it gets faster as you practice, but still. . .
- Test Script framework Integration
- -> Model Portability between tools -> Test Coverage Report -> Effort Savings Report -> ROI Report
- It's the paradigm shift that is most limiting. Those who traditionally write requirements could do the modeling instead, but modeling is hard. It's a different way of thinking. Thus, at this point, modeling is a QA activity. Even though we see a model as a graphical representation of the correct implementation of requirements, models are still developed FROM requirements.
- The models being so tightly coupled to the code, which means that when features change models often have to be updated, and when the system evolves, so must the models. Which us time consuming and error prone.
- Trying to be something more specific than just Testing. It's testing. Not "Model-based". And please stop trying to limit yourself to one model in these questions.
- up-front investment into building a good model when requirements are in a state of flux
- management inertia...

Q25: I see the biggest advantage of MBT as:

Answer Options	Response Count
	55
answered question	55
skipped question	49

Answers:

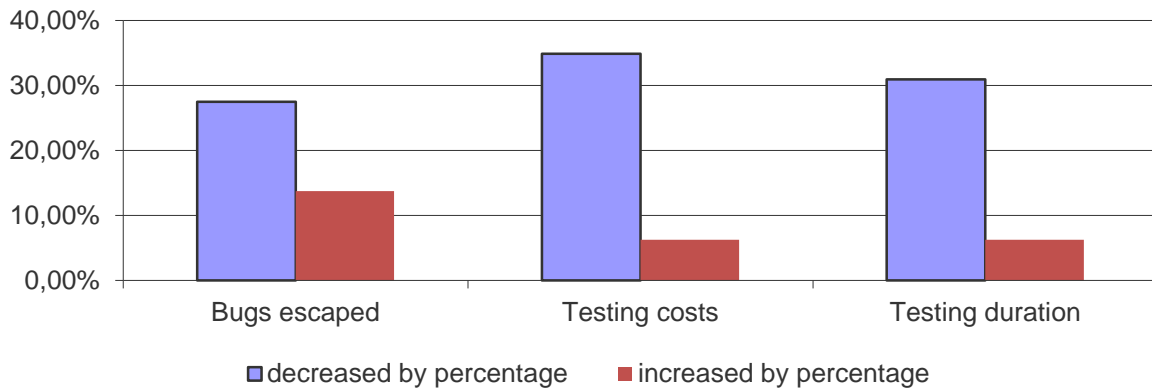
- abstraction level
- Reduce of cost.
- automatic test case generation
- Understanding and test support
- understandability of tests better communication between system engineers and test engineers possibility to generate different test scripts out of one unchanged test model
- uniformity
- efficiency
- The abstraction of complexity
- Handling complexity that leads to improved efficiency, effectiveness and stakeholders satisfaction.
- it structured the testing process
- Cost saving, accuracy
- When faced with complex application functionality, I have seen MBT successfully generate test to thoroughly cover the application paths and find bugs that had been in the product for years (through countless manual regression passes). It can be very beneficial when used appropriately.
- forces the user to think a lot about the system under test test suite maintenance: it is much cheaper/easier to update the model and the regenerate the tests then to modify all the affected test cases manually.
- Discussions between differentr steakholder in an very early project state. Deploying the same picture of the system.
- 1. Abstraction. 2. Automation. leading to: 3. Self-testing.
- regression testing
- In the initial phase of testing also we have the complete design of our module so testing become easy and effective
- Avoid duplication , Single platform and reduce cost and time
- Once the model is createdt you can easily update the behaviour and generate Tests
- Test Case Generation
- Decreased test costs
- Time and test coverage
- makes the tester's world easier
- Nice Tests Coverage for one features. Help to create automatic tests.
- easy to get test coverage
- Being able to run smoke tests on any build, on many devices, secure a certain quality level on builds with respect to key metrics and functionality.
- complex scenarios
- Increase test coverage
- higher test throughput, higher degree of verification and validation
- Reducing the total ROI for testing & improved test collateral & test coverage
- As we create a more robust model it becomes more efficient to provide testing services since the test generator speeds up all testing process
- A more productive test proces
- Getting structure in bad documentation by using models to review the testbasis.
- maintenance fo a few models instead of hundreads/thousands of test cases
- the possibility to perform an early validation of the test basis through modeling
- Reuse of models and model elements

- Enhanced agility of testing in response to changes, effective and efficient test design and test automation.
- Work in a global team
- variation in regression testing
- Organize and see the end result of testing.
- Effective testing
- -> Model Reusability -> Better Test cases/scenarios generation -> End to End Test Design & Test Execution Automation
- Long sequence tests; automatically testing scenarios that no tester would consider doing manually;
- If it is possible to reason about a system on an abstract level.
- It's already being used by all testers. Just stop trying to call it something else than just testing.
- having a good model that drives testing
- effectively testing and 'knowing' the impact of testing...
- fast improvement of coverage
- high coverage and automation
- test case generation and support for coverage metrics
- Only practical way to objectively assess system reliability.
- Control on coverage Capitalize the knowledge in MBT models
- improved test effectiveness ("better tests").
- 1) Automated test case generation. 2) Facilitating the maintenance of the test automation process, thus increasing the return of investment of test automation.
- Improvements in communication and reviews (additional stakeholders) Effort reduction (reuse, avoided redundancies, tool support) Knowledge of test coverage and test quality Scalable test coverage and test quality that can be adopted effectively based on the objectives of the actual test phase

**Q26: Please provide your best estimate of the effects of MBT.
If not applicable or unsure, leave blank.**

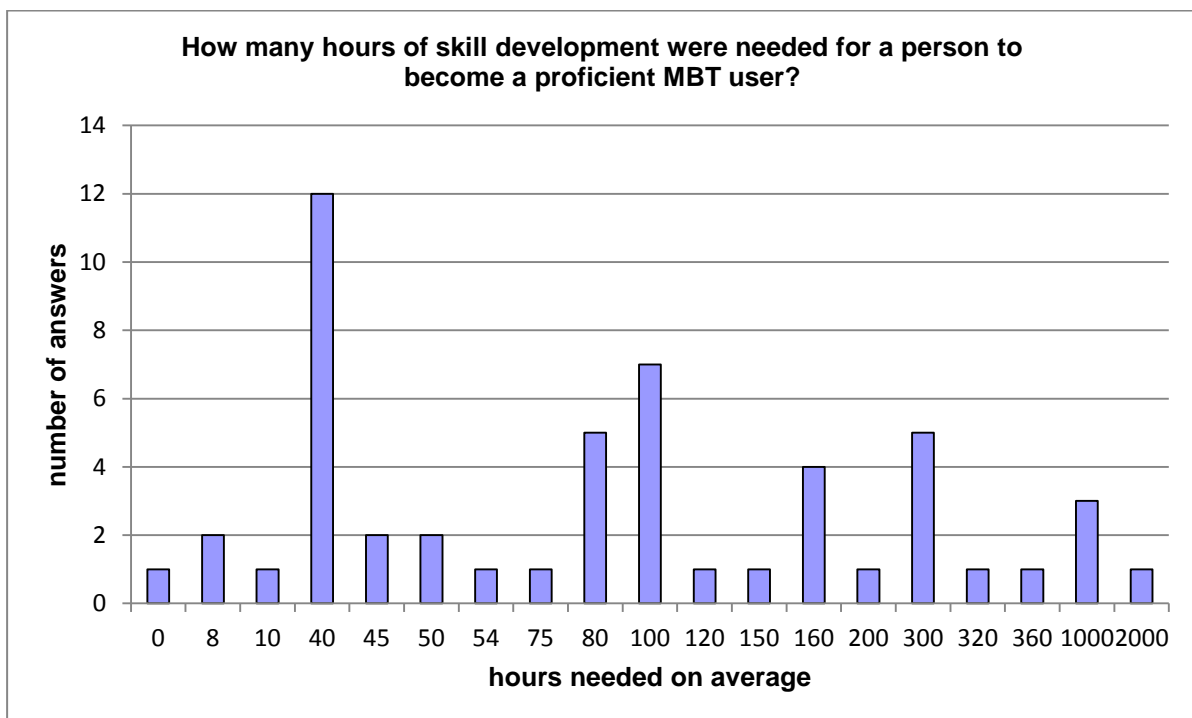
Answer Options	Response Average	Response Total	Response Count
Bugs escaped decreased by percentage:	27,50	330	12
Bugs escaped increased by percentage:	13,75	55	4
Overall testing costs decreased by percentage:	34,89	663	19
Overall testing costs increased by percentage:	6,25	25	4
Overall testing duration decreased by percentage:	30,93	464	15
Overall testing duration increased by percentage:	6,25	25	4
answered question			21
skipped question			83

Best estimate of the effects of MBT (21 answers)



Q27: On average, about how many hours of skill development were needed for a person to become a proficient MBT user?

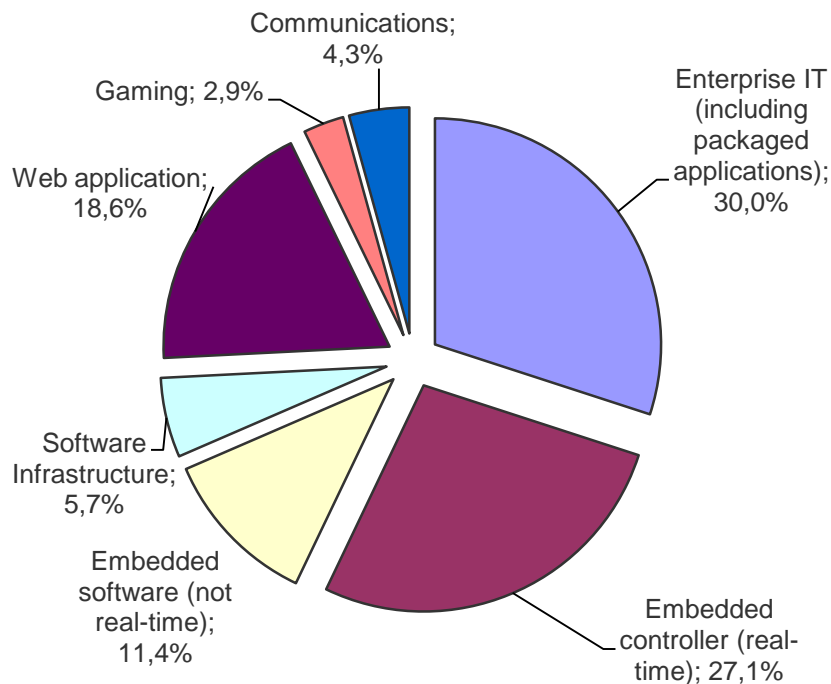
Answer Options	Response Average	Response Median
nn hours	196,4	80
<i>answered question</i>		52
<i>skipped question</i>		52
minimum value		0
maximum value		2000



Q28: What is the general application domain of the system under test?

Answer Options	Response Percent	Response Count
Enterprise IT (including packaged applications)	30,0%	21
Embedded controller (real-time)	27,1%	19
Embedded software (not real-time)	11,4%	8
Software Infrastructure	5,7%	4
Web application	18,6%	13
Gaming	2,9%	2
Communications	4,3%	3
Others:		6
answered question		70
skipped question		34

What is the general application domain of the system under test?

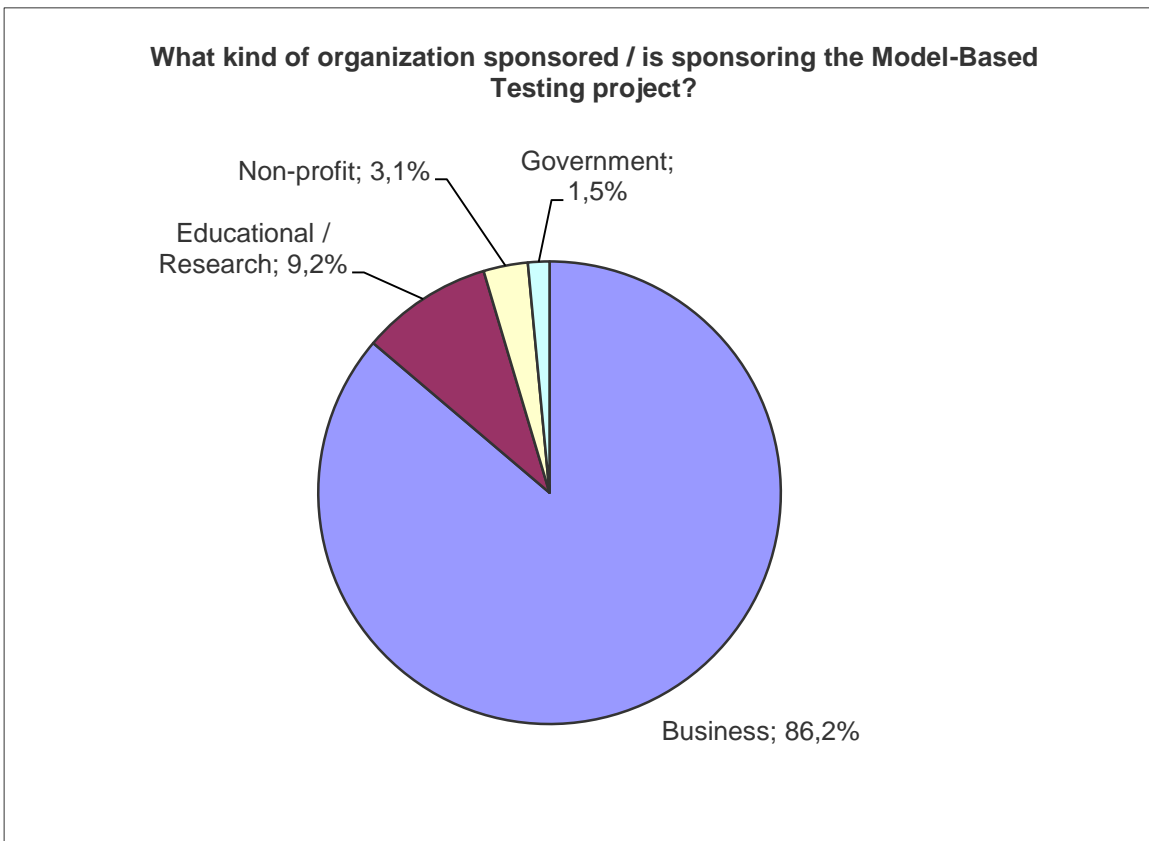


Others:

- telecom
- infrastructure for distributed and/or real-time systems, database applications, server/client systems
- It spans over multiple domain: SOA embedded systems desktop applications distributed systems
- but also Enterprise IT and Software Infrastructure
- There is more than one system to test. So you could say: all of the above".
- education,

Q29: What kind of organization sponsored / is sponsoring the Model-Based Testing project?

Answer Options	Response Percent	Response Count
Business	86,2%	56
Educational / Research	9,2%	6
Non-profit	3,1%	2
Government	1,5%	1
Other (please specify)		2
answered question		65
skipped question		39

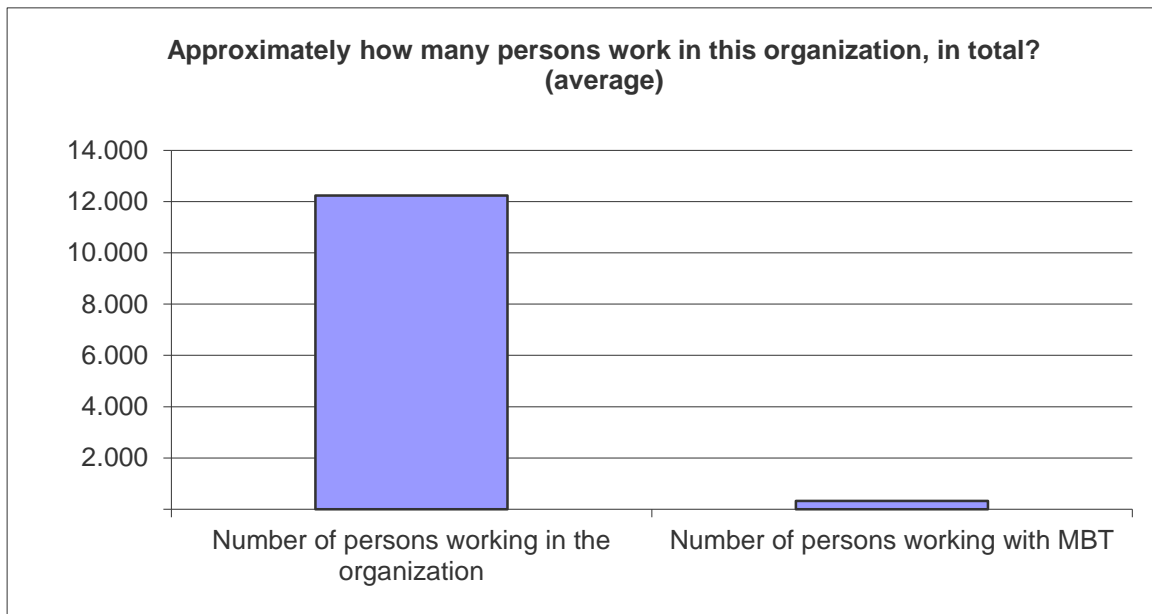


Others:

- Engineering
- no sponsoring at all, part of real projects

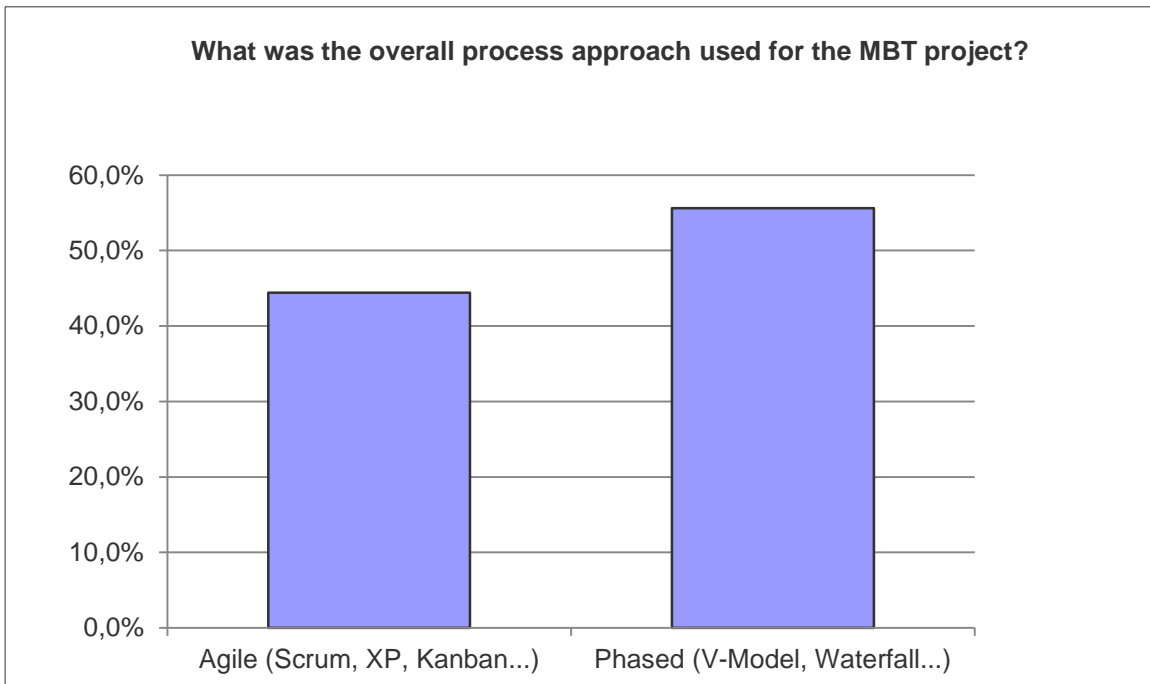
Q30: Approximately how many persons work in this organization, in total?

Answer Options	Response Average	Response Median	Response Count
Number of persons working in the organization	12.236	15	51
Number of persons working with MBT	325	5	47
answered question			53
skipped question			51
minimum value (total)			3
maximum value (total)			3000000
minimum value (MBT)			1
maximum value (MBT)			200



Q31: What was the overall process approach used for the MBT project?

Answer Options	Response Percent	Response Count
Agile (Scrum, XP, Kanban...)	44,4%	28
Phased (V-Model, Waterfall...)	55,6%	35
Other (please comment)		9
answered question		63
skipped question		41

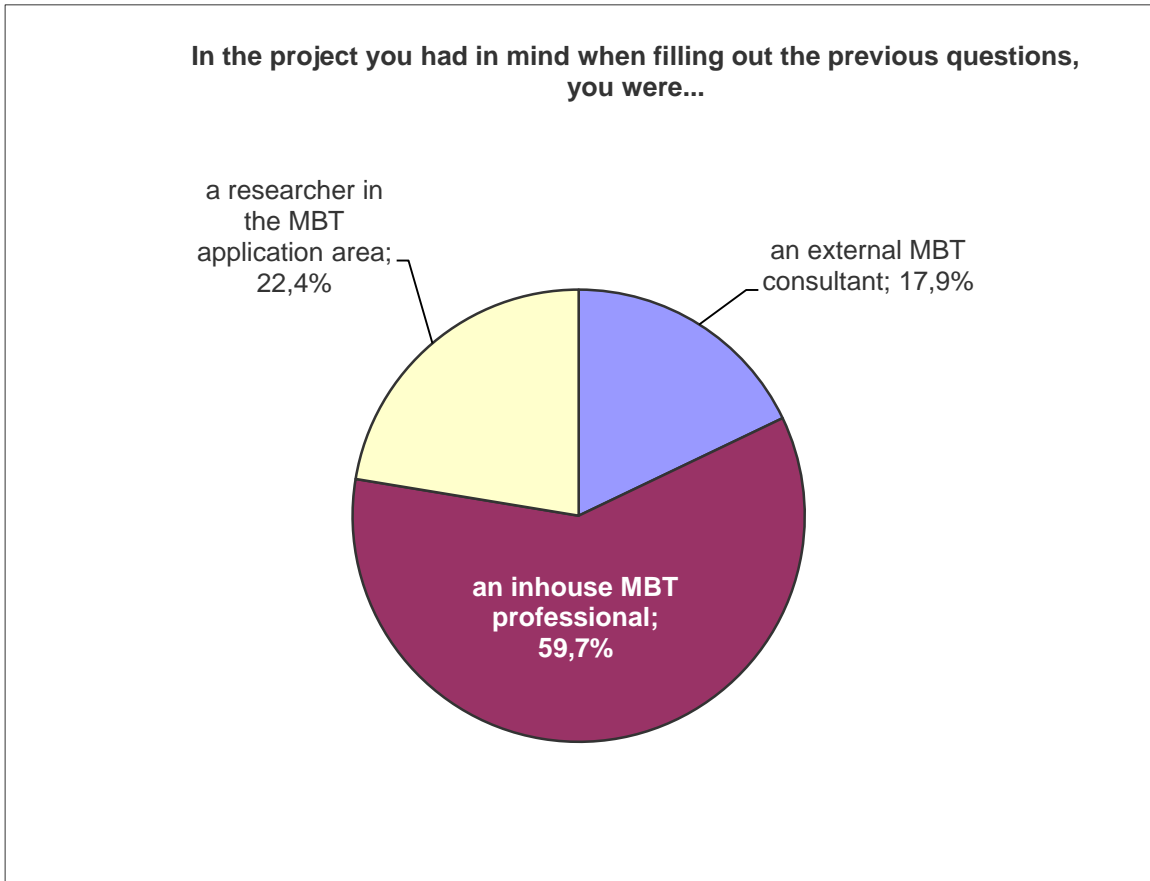


Others:

- both
- Am not sure about Agile
- incremental, iterative which does not mean Agile
- using executable specifications
- MBT is difficult to fit in to Agile approach.
- Like most projects I know, it just receives a name if "process methodologists" insist it should be called something other than "software development".
- hybrid Agile == Waterfall + Agile == Wagile
- Spiral
- model-driven development

Q32: In the project you had in mind when filling out the previous questions, you were...

Answer Options	Response Percent	Response Count
an external MBT consultant	17,9%	12
an inhouse MBT professional	59,7%	40
a researcher in the MBT application area	22,4%	15
Other (please specify)		2
answered question		67
skipped question		37



Others:

- Test manager
- A professional tester.