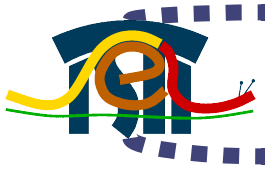
The 17th CREST Open Workshop
Software Testing and Verification

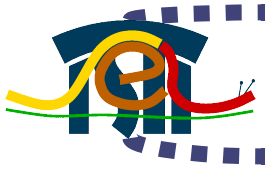
# Recent trends in *software testing research*: an unsystematic (road)mapping study

Antonia Bertolino



# Agenda

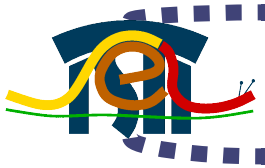
- Introduction
- What is software testing
- Background
- Re-check/update to 2007's roadmap
- Roadmap 2012
- Testing & Verification
- Conclusions



# Agenda

- Introduction (focus)
- What is software testing (my own definition)
- Background (FOSE 2007 roadmap)
- Re-check/update to 2007's roadmap (an unsystematic mapping study)
- Roadmap 2012 (weighted and augmented)
- Testing & Verification (discussion hints)
- Conclusions (quick!)

# Introduction



# Who am I: antonia.bertolino at ISTI.CNR.IT

- Improving and automating structural testing
- Precise dataflow-based test criteria (to prevent untestedness)
- Operational testing (for reliability evaluation)
- Testability
- IEEE SWEBOK
- SA-based testing
- UML-based testing
- CB testing
- SOA testing
- XML-based testing

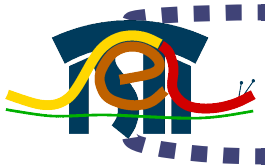
http://www.isti.cnr.it/People/A.Bertolino

# Focus: overview of software testing


**Testing software artifacts using various approaches and against different targets ...**

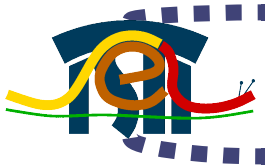
...from an academic perspective: → ST research

- Why do we need software testing research?
- What are current open issues, proposed approaches and hottest topics?
- How can ST research benefit from formal methods/verification research?



# Relevance of software testing for practitioners

- Testing remains the widest industrial approach to Verification&Validation
- Notwithstanding the best efforts in formalization, rigour and advanced design techniques, faults will always enter the software product
- Testing is essential for demonstrating software behavior
  - Testing is tricky, smart testers are wanted!



# Relevance of software testing for researchers:

ST still remains expensive, difficult and largely ad hoc
(as opposed to systematic and automatic):

- quotes for relative cost of testing still similar to 20 years ago or worse
- sophisticated support tools require the background of a PhD in formal methods plus intensive dedicated training
- ST resources are constrained : we need effective ST technology and tools to decrease costs

ST is unpredictable

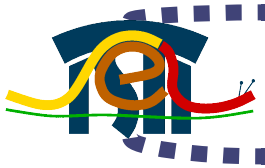
- we need theory and (empirical) insights to associate ST techniques to effects

Evolution of software systems:

- novel technologies and novel challenges: ST techniques and processes that are good today may not be good tomorrow
- growing complexity and pervasiveness of software systems
- Functional + non-functional requirements intertwined

Gap between what is "preached" and what is practiced:

- is this testing-peculiar?
- why the gap, and what can we do?



# Research in ST

- Increasing awareness and interest: from niche research to pervasive
- Many focused Conferences and Workshops: ICST, ISSTA, Automation of Software Test (AST); Advances in Model Based Testing (A-MOST); Testing - Academic & Industrial Conference Practice And Research Techniques (TAIC-PART); Domain-Specific Approaches to Software Test Automation (DoSTA); …
- Growing percentage of papers at main SE conferences

# What is software testing?

# What testing is not

- **I've searched hard for defects in this program, found a lot of them, and repaired them. I can't find any more, so I'm confident there aren't any.**

The fisher analogy:

- **I've caught a lot of fish in this lake, but I fished all day today without a bite, so there aren't any more.**

- NB: Quantitatively, a day's fishing probes far more of a large lake than a year's testing does of the input domain of even a trivial program.

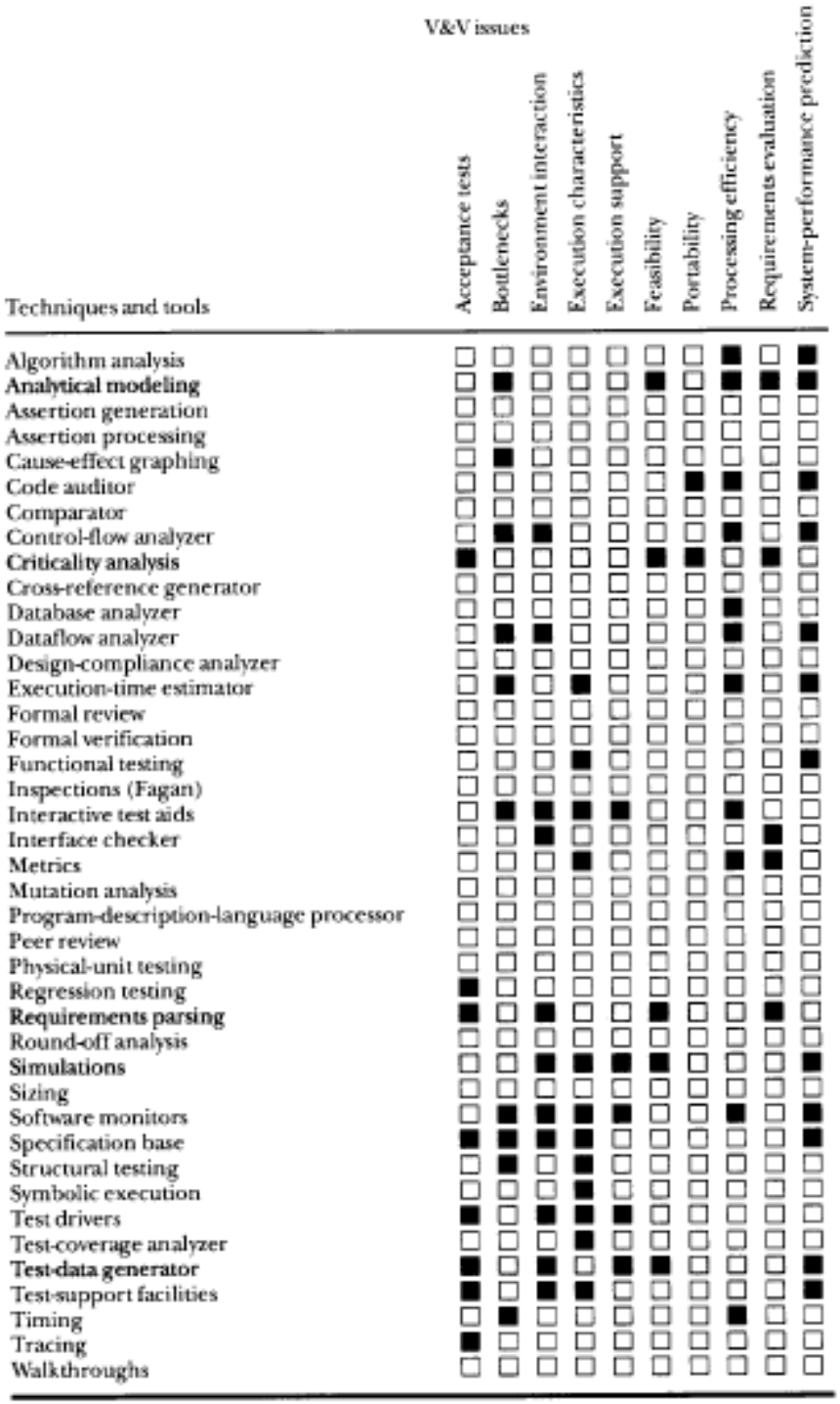
COW: Hamlet, D. 1994. Foundations of software testing: dependability theory. In *Proc. 2nd ACM SIGSOFT FSE 1994.*

# An all-inclusive definition

Software testing consists of

- the dynamic verification of the behavior of a program
- on a finite set of test cases
- suitably selected from the (in practice infinite) input domain
- against the expected behavior

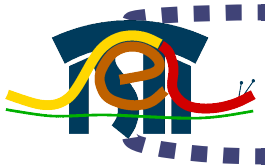
A. Bertolino, "Software Testing", Chapter 5, in
IEEE Guide to the SWEBOK (2001, stoneman v.), www.swebok.org




# Testing vs. Other Approaches

- Many V&V techniques and tools
- that span over different phases and address different objectives

D.R. Wallace and R.U. Fujii, “Software Verification and Validation: An Overview,” IEEE Software 6 (3), 1989



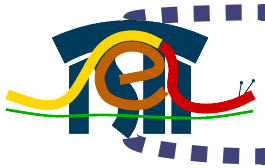
# Testing vs. Other Approaches

- Where is the boundary between testing and other (analysis/verification) methods?

What distinguishes software testing from "other" approaches is execution of the SUT. This requires the ability to:

- launch the tests on some executable version (traceability problem), and
- analyse (observe and evaluate) the results

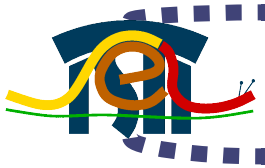
- Everyone would agree that testing and other V&V methods should complement/support each other: open question is how?



# Testing vs. Verification

- This workshop:
  - … two complementary technique sets: while verification seeks to ensure freedom from certain classes of faults, testing seeks to identify those that may yet remain …
    - More later….

# Background





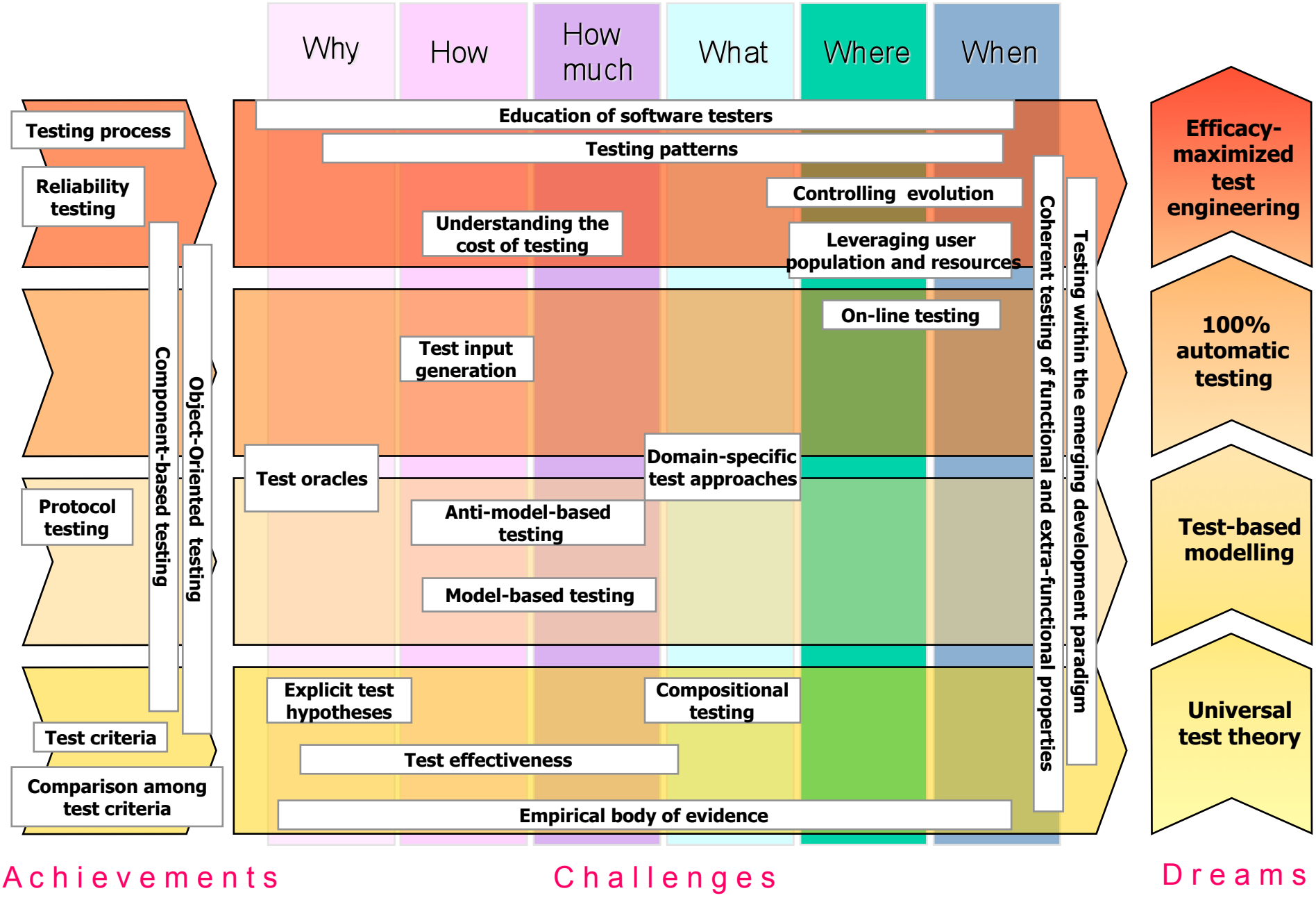
# Software Testing Research: Achievements, Challenges, Dreams

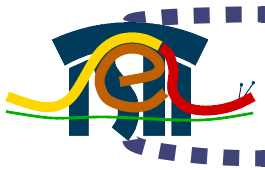
In ACM Future of Software Engineering (companion volume of ICSE 07 Proc.)

29th Int. Conference on Software Engineering ICSE 2007 Minneapolis (MN) USA May 19 - 27, 2007

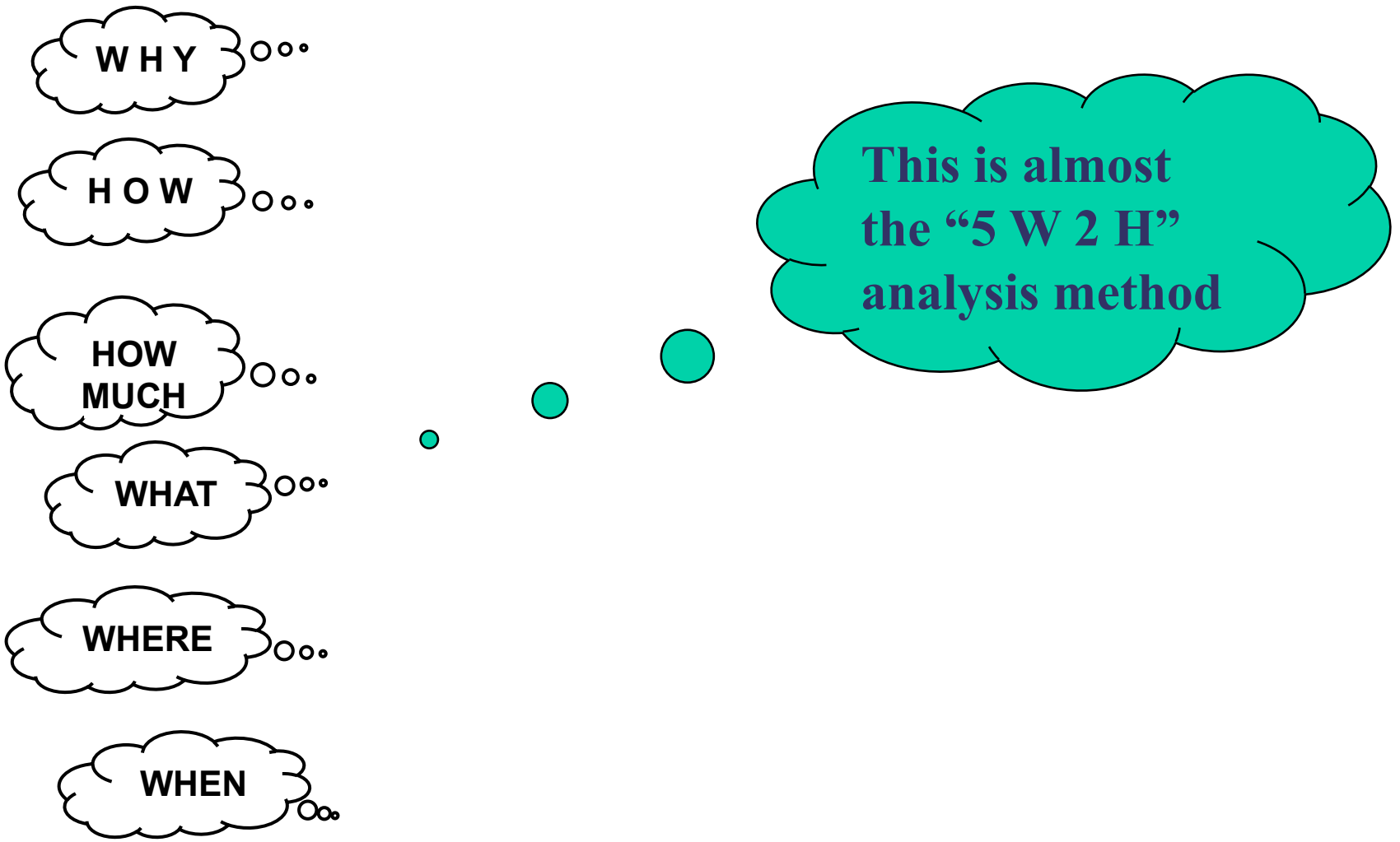
- Google scholar: 239 citations
- Among the Top 10 Downloads from whole ACM's DL Sept. 2007

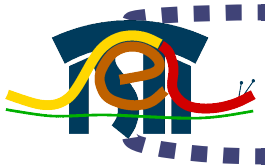
# Software testing research roadmap





# Testing research scopes





# Testing research scopes


Objectives for testing can be different

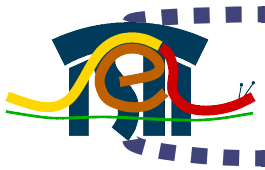

Many techniques and tools


When is it enough? (the stopping rule)


The SUT (varies along the life cycle)

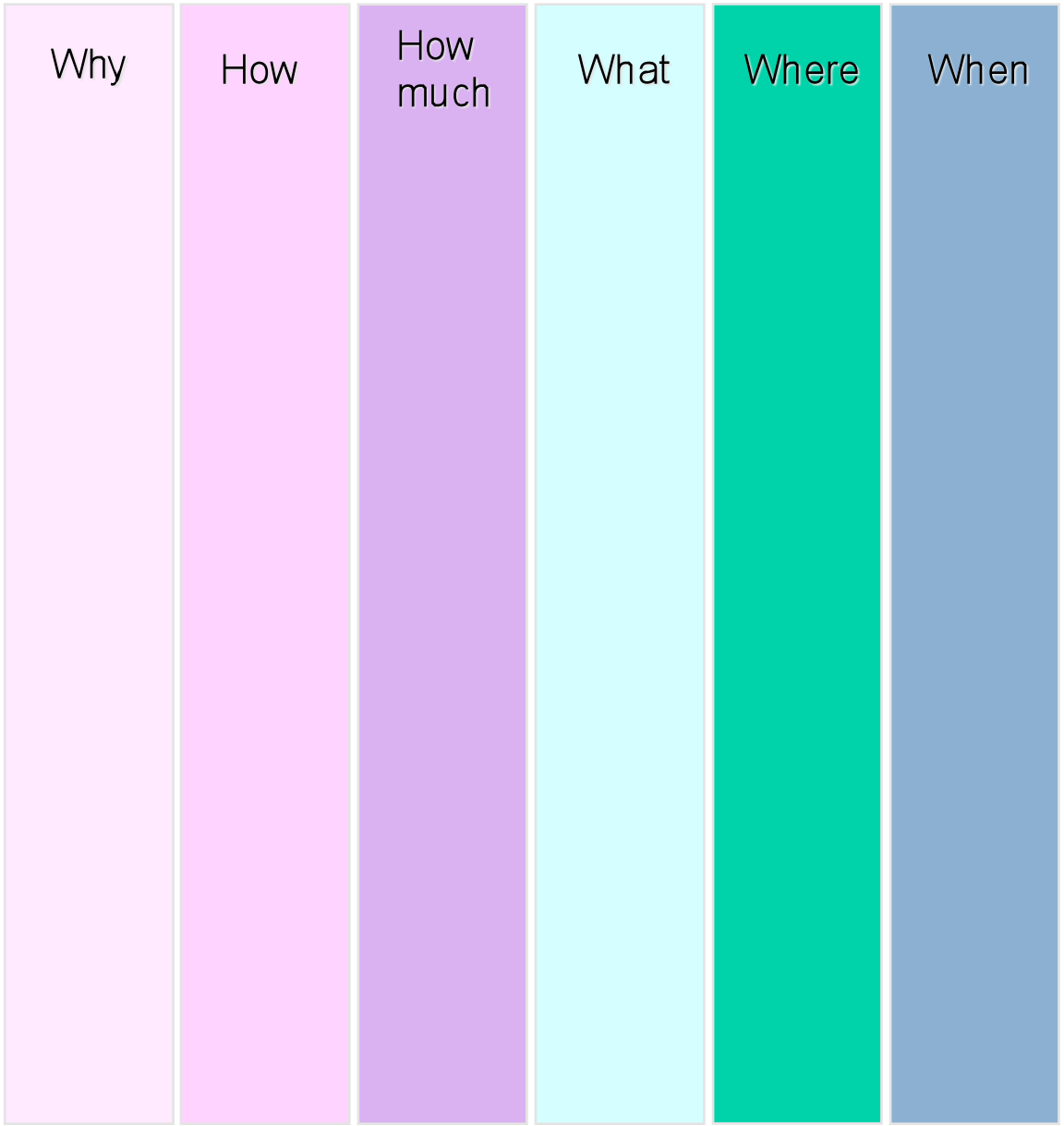

The test environment (host vs target)


Test process; off-line – on-line



# Testing research scopes

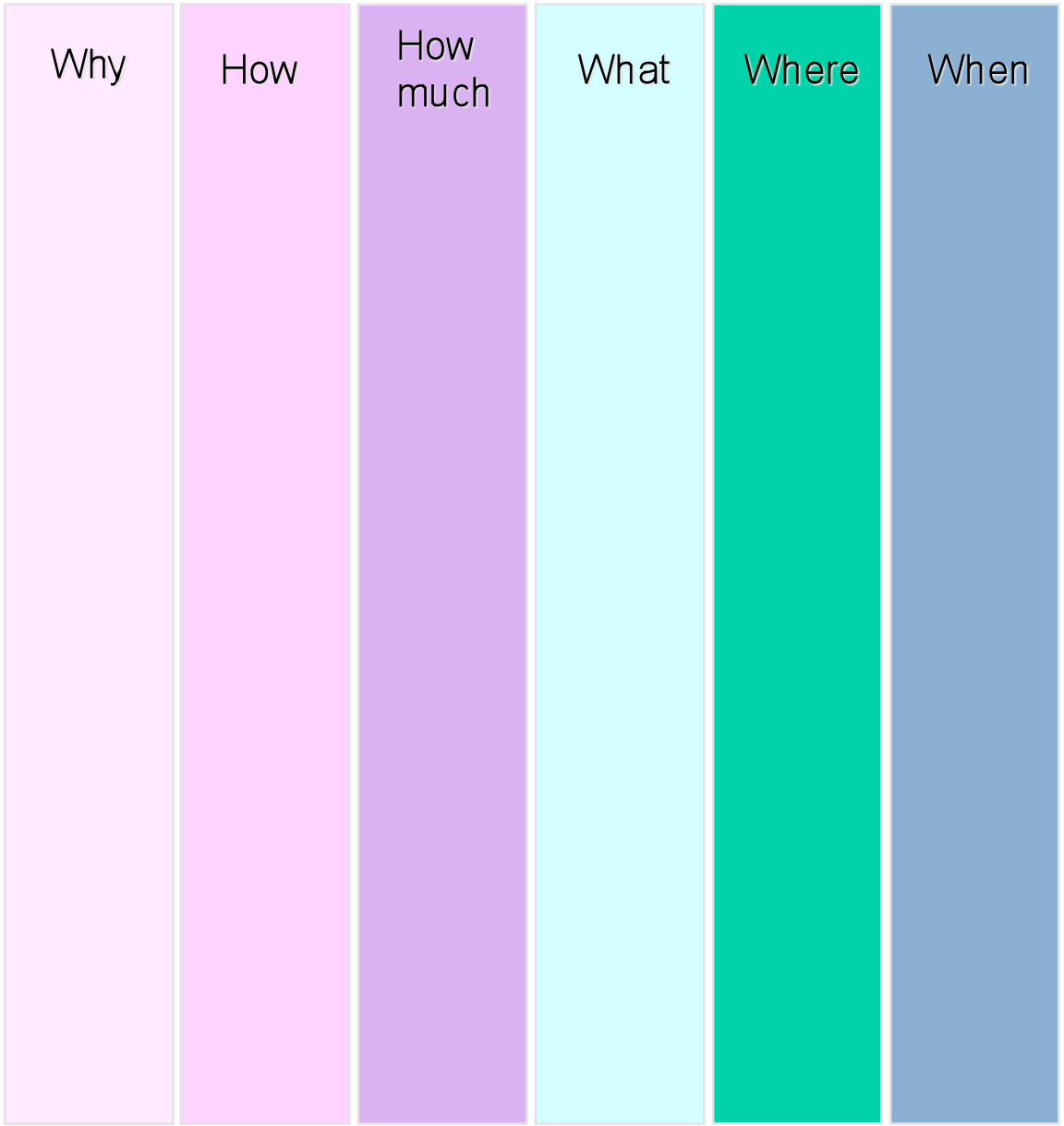
| | |
|---|---|
| WHY | Objectives for testing can be different |
| HOW | Many techniques and tools |
| HOW MUCH | When is it enough? (the stopping rule) |
| WHAT | The SUT (varies along the life cycle) |
| WHERE | The test environment (host vs target) |
| WHEN | Test process; off-line – on-line |




Achievements

Challenges

Dreams




Universal test theory

Achievements

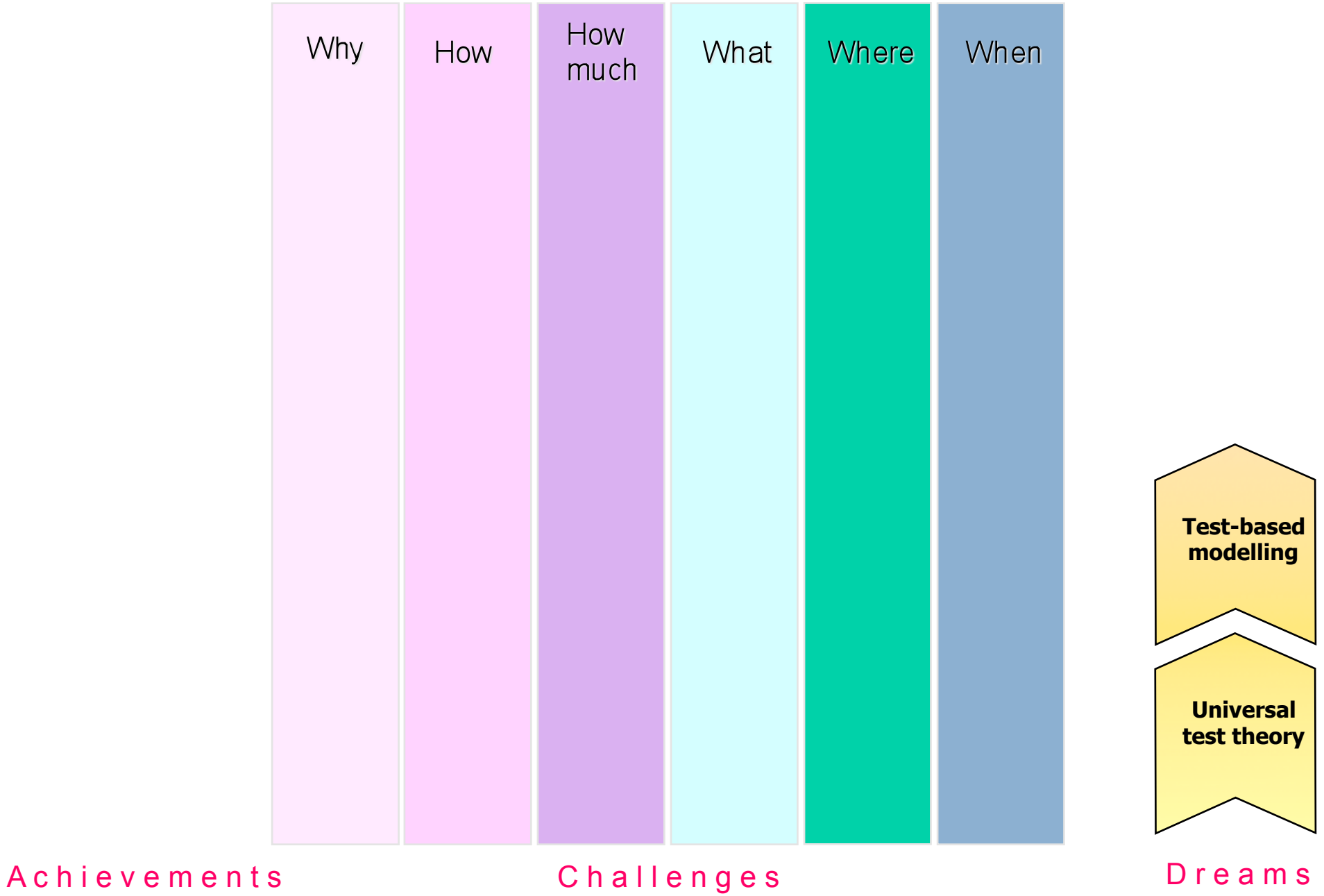
Challenges

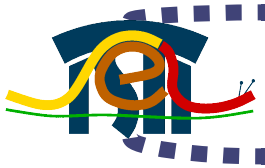
Dreams

# Universal Test Theory

- A comprehensive support theory that provides useful explanation and guidance to practice:
  - **What can I expect by applying this test technique to this situation?**
  - **Which techniques should I choose in the present circumstances?**
  - ….
- Seminal work in early 70's: mostly a negative theory, exemplified by Dijkstra's aphorism that testing can only show the presence of errors, but never their absence

# Software testing research roadmap 2007





# Test-based Modeling

From: MODEL → TEST

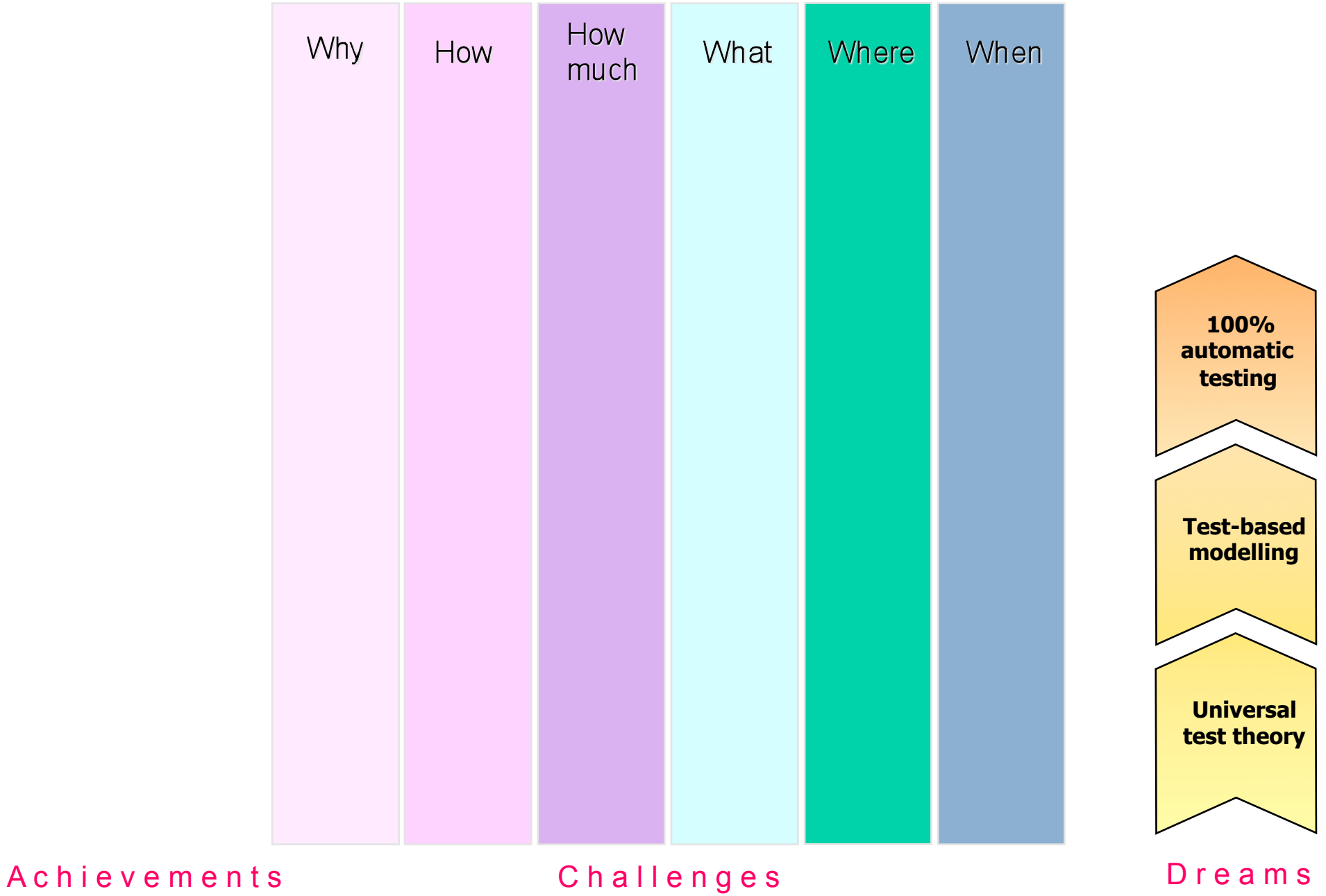
To: MODEL ← TEST

i.e., think ahead of models and instruments that support effective testing

Would build on notions from:

- "design for testability" – Built-in test
- design-by-contract (assertion-based programming)

# Software testing research roadmap 2007



# 100% Automatic Testing

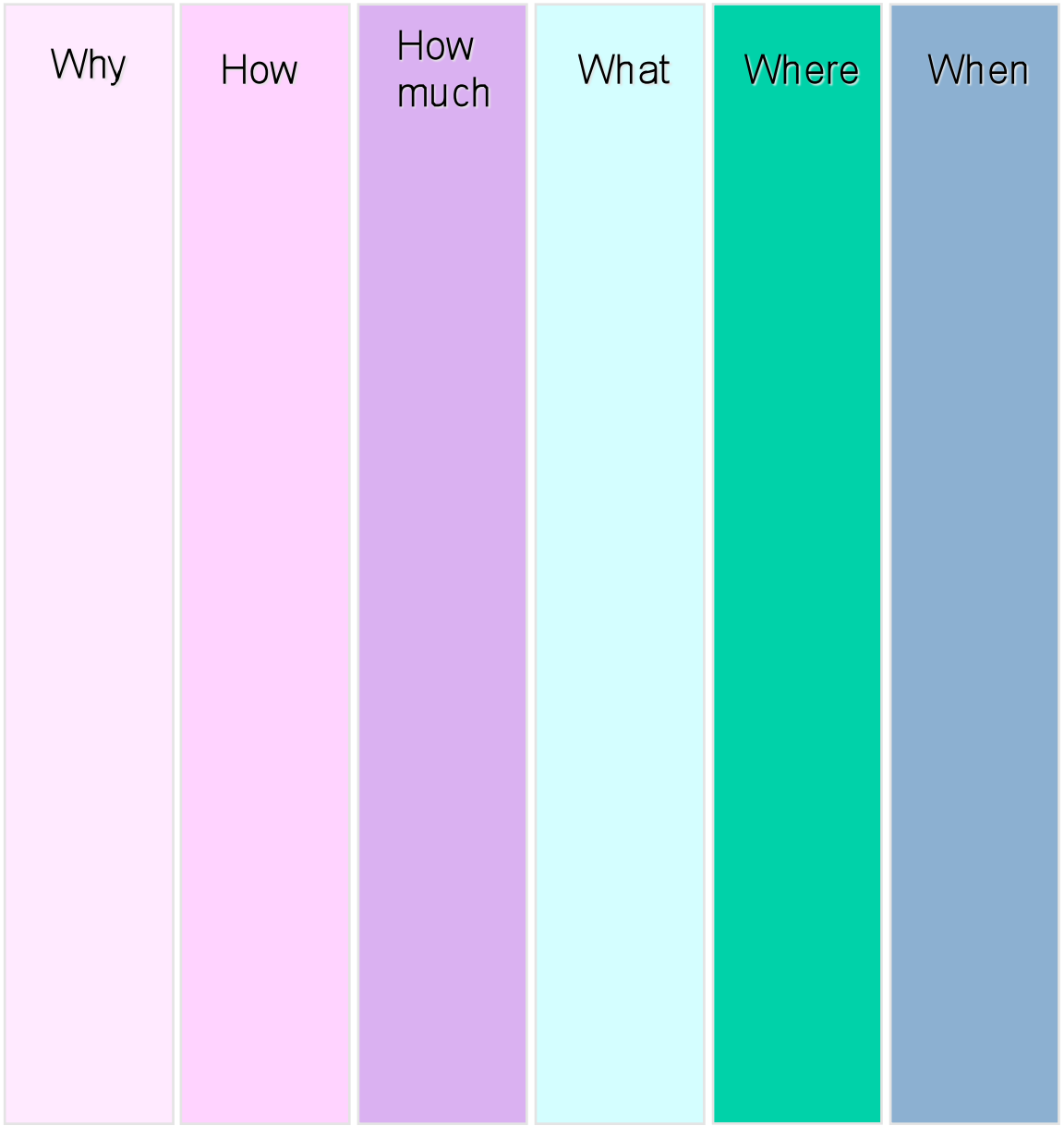
Automation of

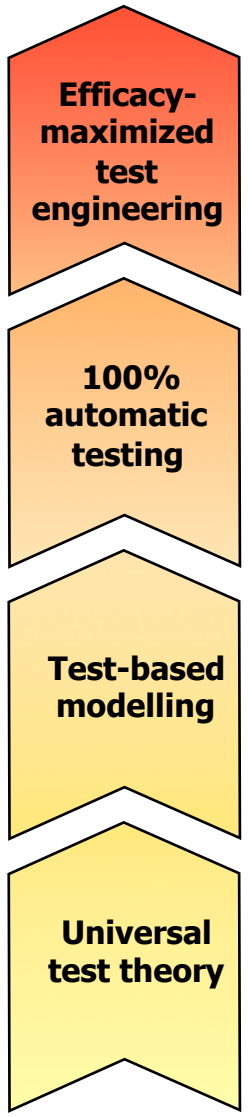
- ✓ test case generation
- ✓ test scaffolding
- ✓ test management: planning, logging, analysis

Progress from the combination of (strengths of) different approaches (especially at unit level), e.g.,

- dynamic analyses&random
- XUNIT & symbolic execution

# Software testing research roadmap 2007

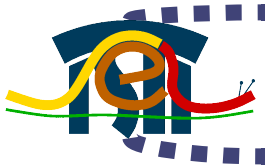



Achievements

Challenges

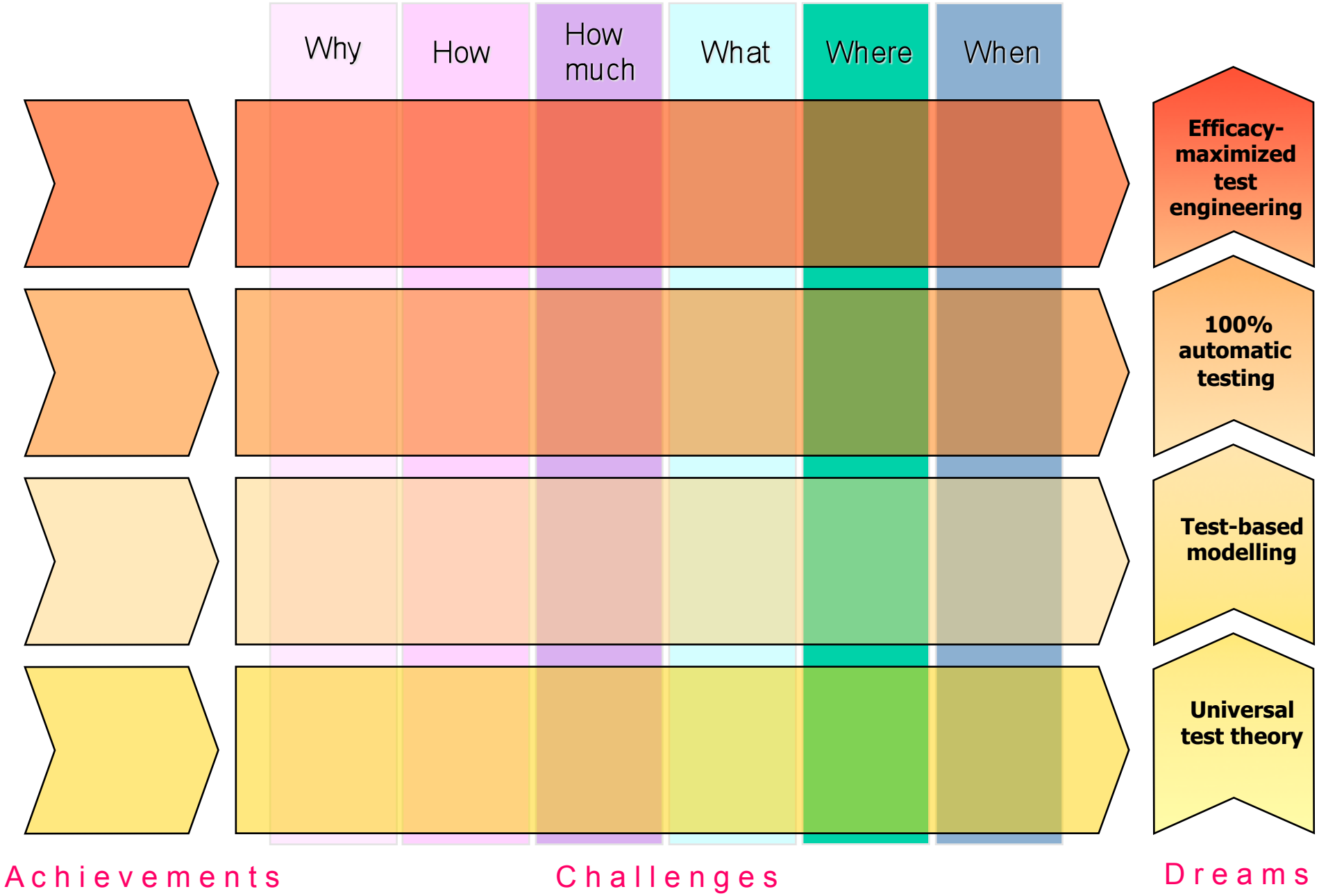
Dreams



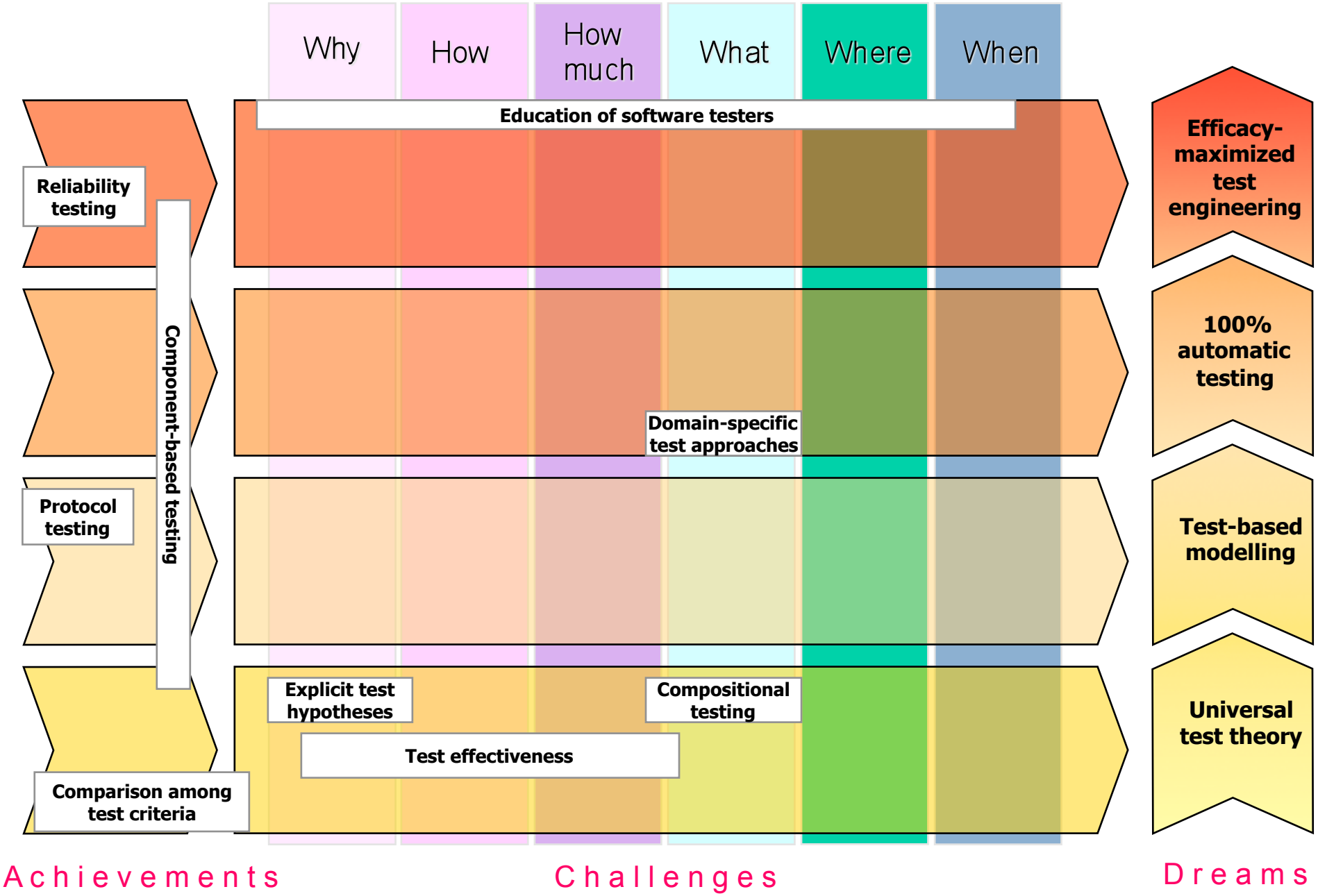
# Efficacy-maximized Test Engineering

- "Efficacy-maximized" as meant to include notions of efficiency/cost and effectiveness/ results
- Global process improvement
- Promoting the transfer of testing research advances from state-of-art to state-of-practice
- Mastering the growing complexity of software: pervasiveness, dynamicity, distribution, heterogeneity, large scale, …

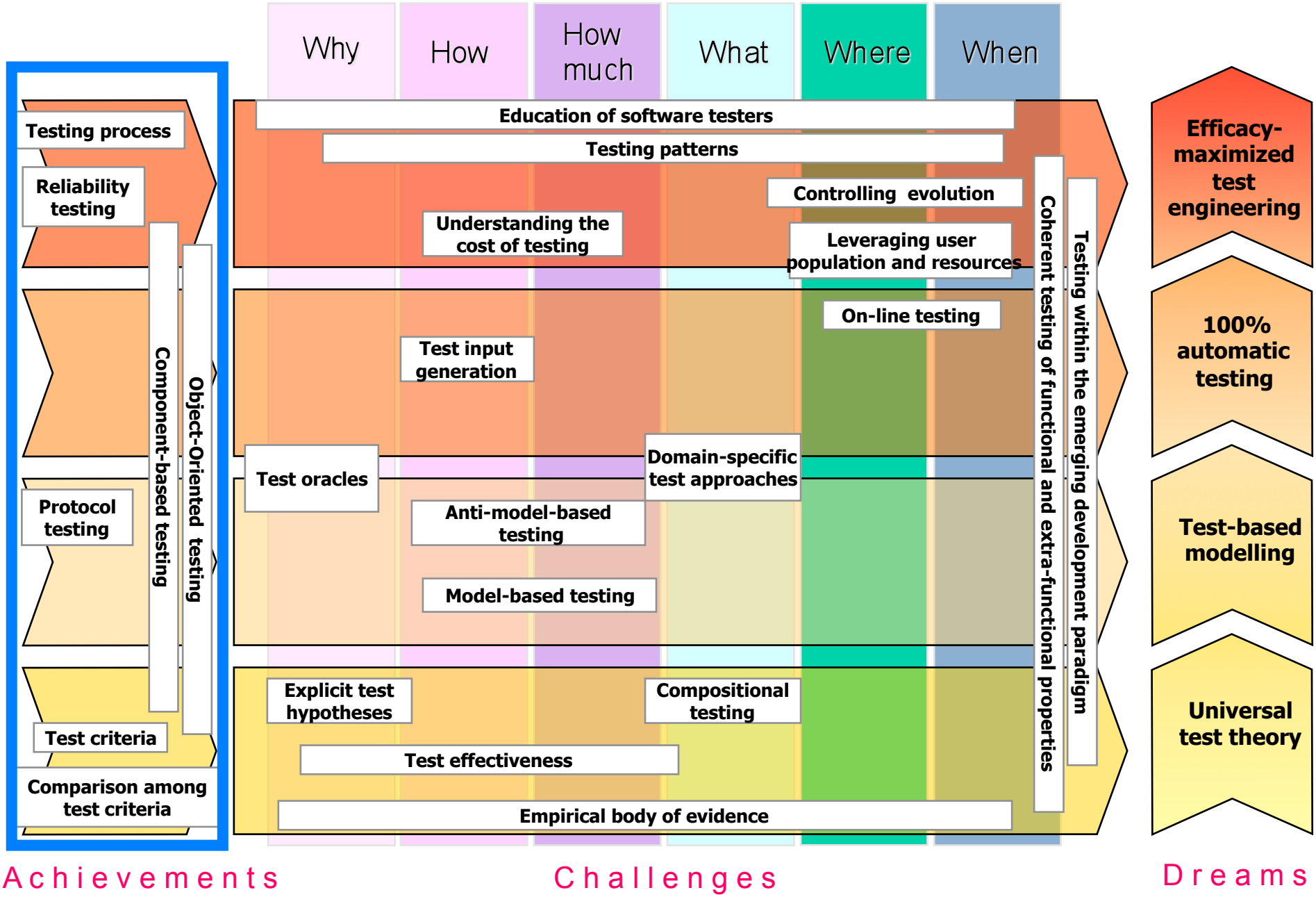
# Software testing research roadmap 2007

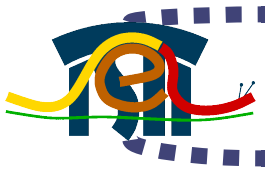


# Software testing research roadmap 2007



# Software testing research roadmap 2007



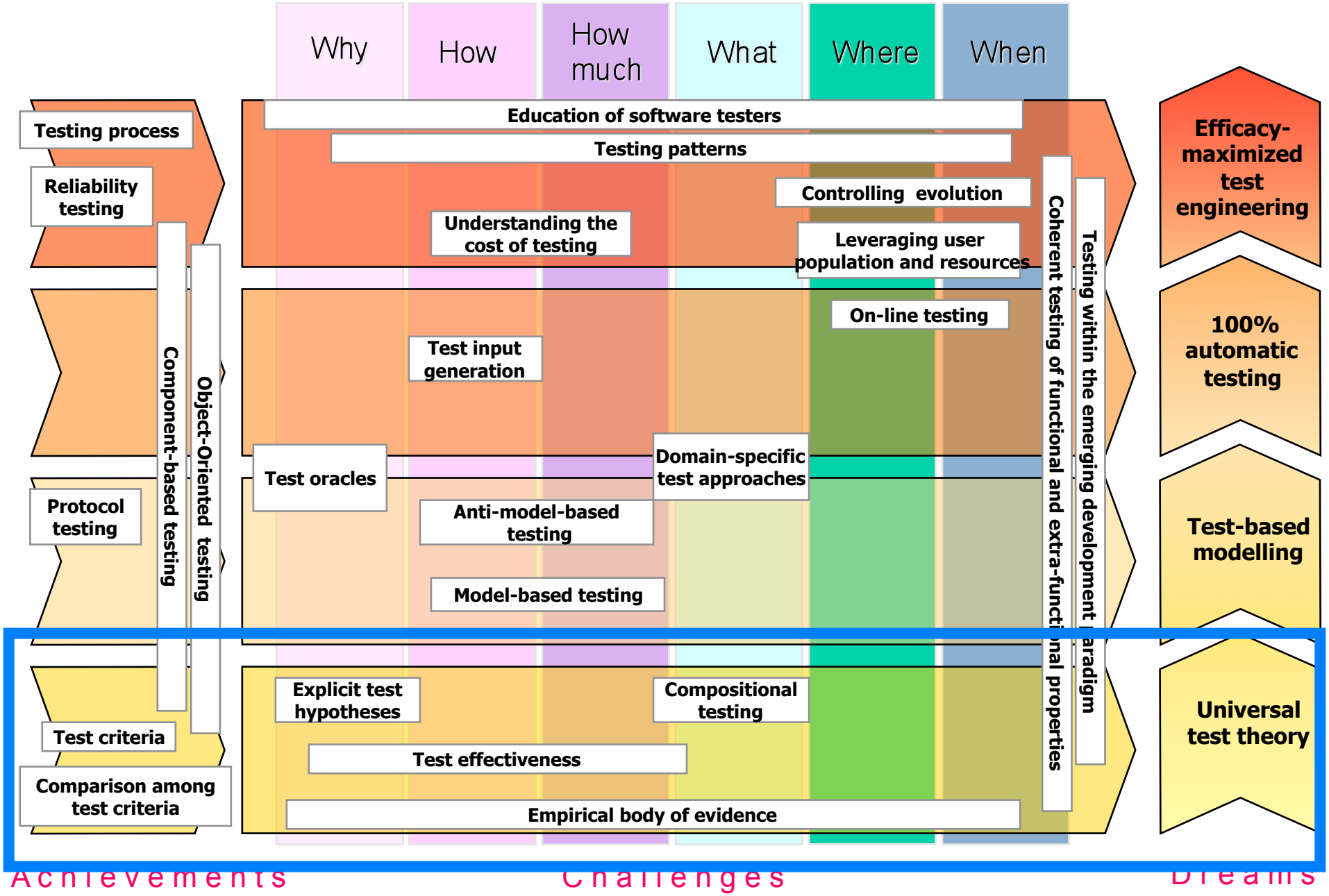


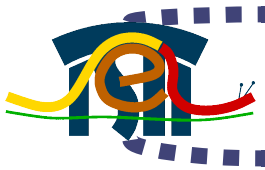
# Achievements

- Software testing discipline originates in the early 70's***
- Several strong achievements, e.g. we know about:
  - Test selection/adequacy criteria
  - Comparison of their thoroughness and effectiveness
  - Test process
  - Reliability testing
  - Protocol testing
  - Object-oriented testing
  - Component-based testing

*** See e.g. my keynote: " A guided tour of four decades of a software testing discipline", 34th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA 2008) , September 3-5, Parma, Italy , IEEE CS 2008.

# Software testing research roadmap 2007

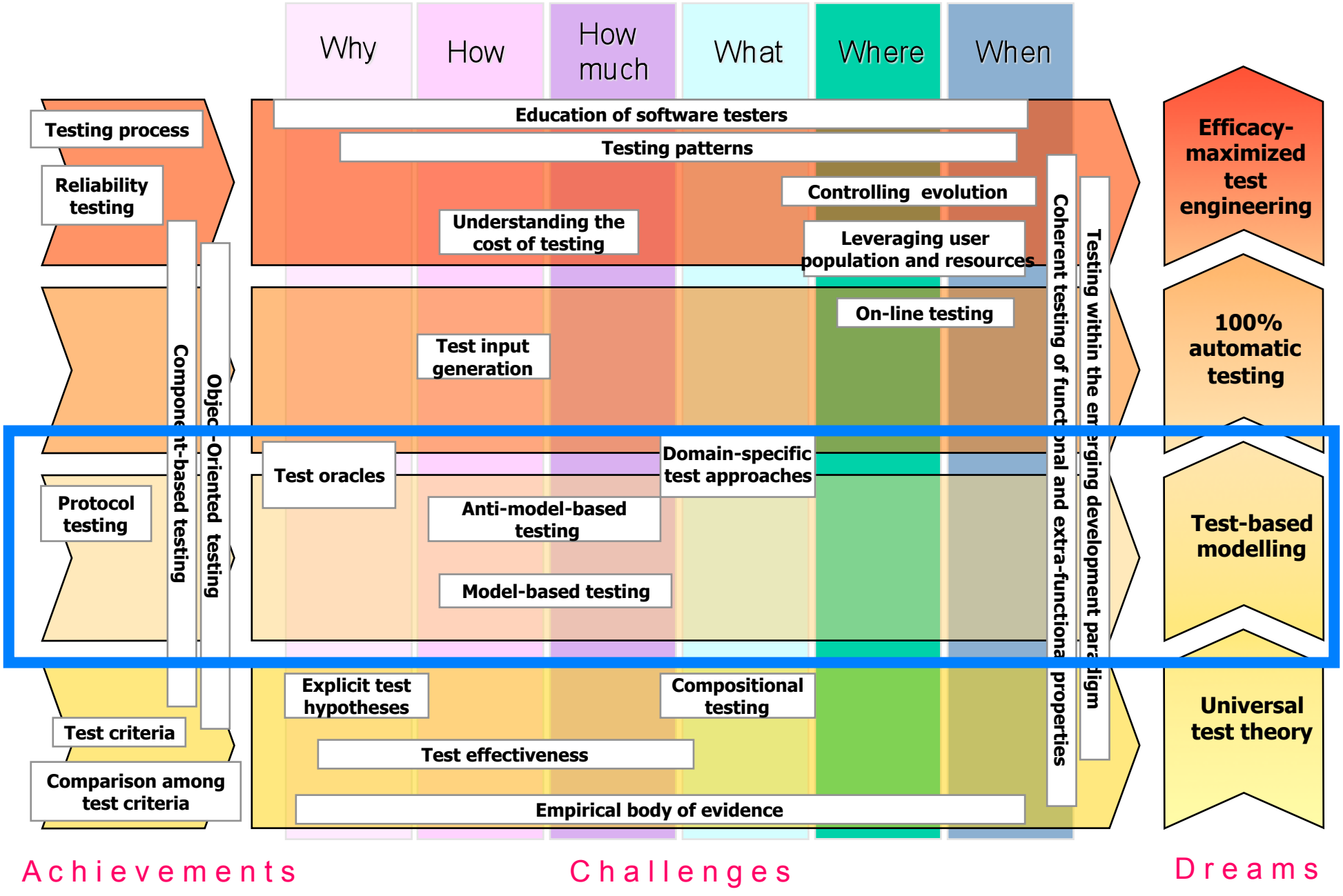


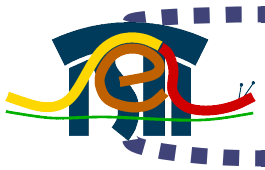


# Challenges to Universal Test Theory

- Explicit test hypotheses (similar to fault model)
  - define the assumptions made behind a test method
- Test effectiveness
  - understand by analysis or experiment the effects of a techniques (which kind of fault, relative effectiveness), so to be able to choose among them
- Compositional testing
  - how to partition the test of a composite system among its components, or viceversa infer conclusions for the whole from the test results of the components
- Empirical body of evidence
  - learn by experiments and build test benchmarks
- ...

# Software testing research roadmap 2007

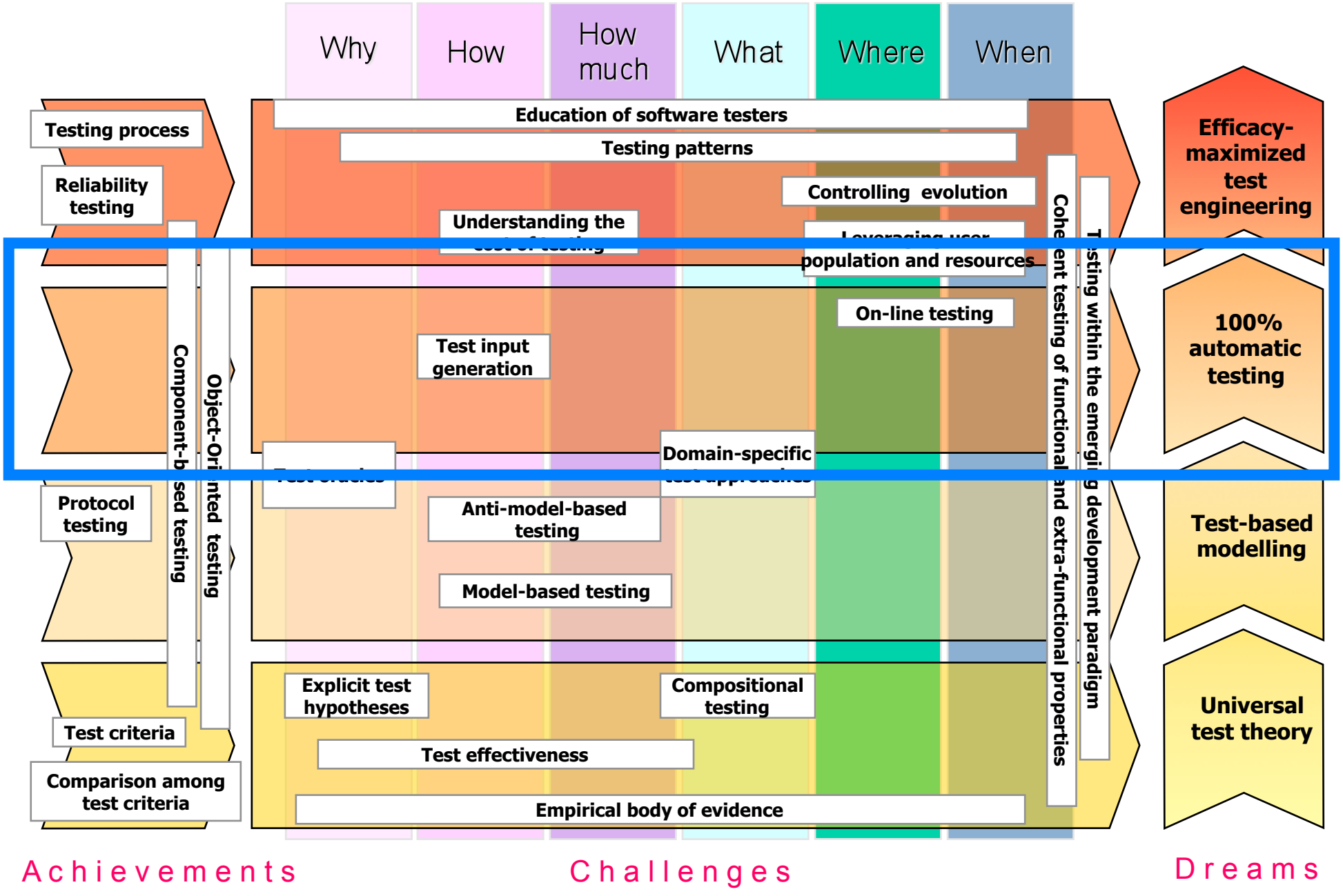


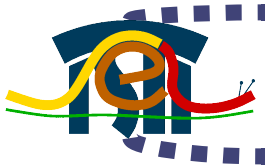


# Challenges to Test-based Modelling

- Model-based testing (MBT)
  - "Back-to-the-future" (e.g., FSM-based testing) , but finally with realistic ambition of industrial take-up
- Anti-model-based testing
  - not always models are available, in such cases they can be reconstructed from test results
- Test oracles
  - neglected issue, models in MBT can be also used as oracles, how to deal with partiality and abstraction?
- ...

# Software testing research roadmap 2007



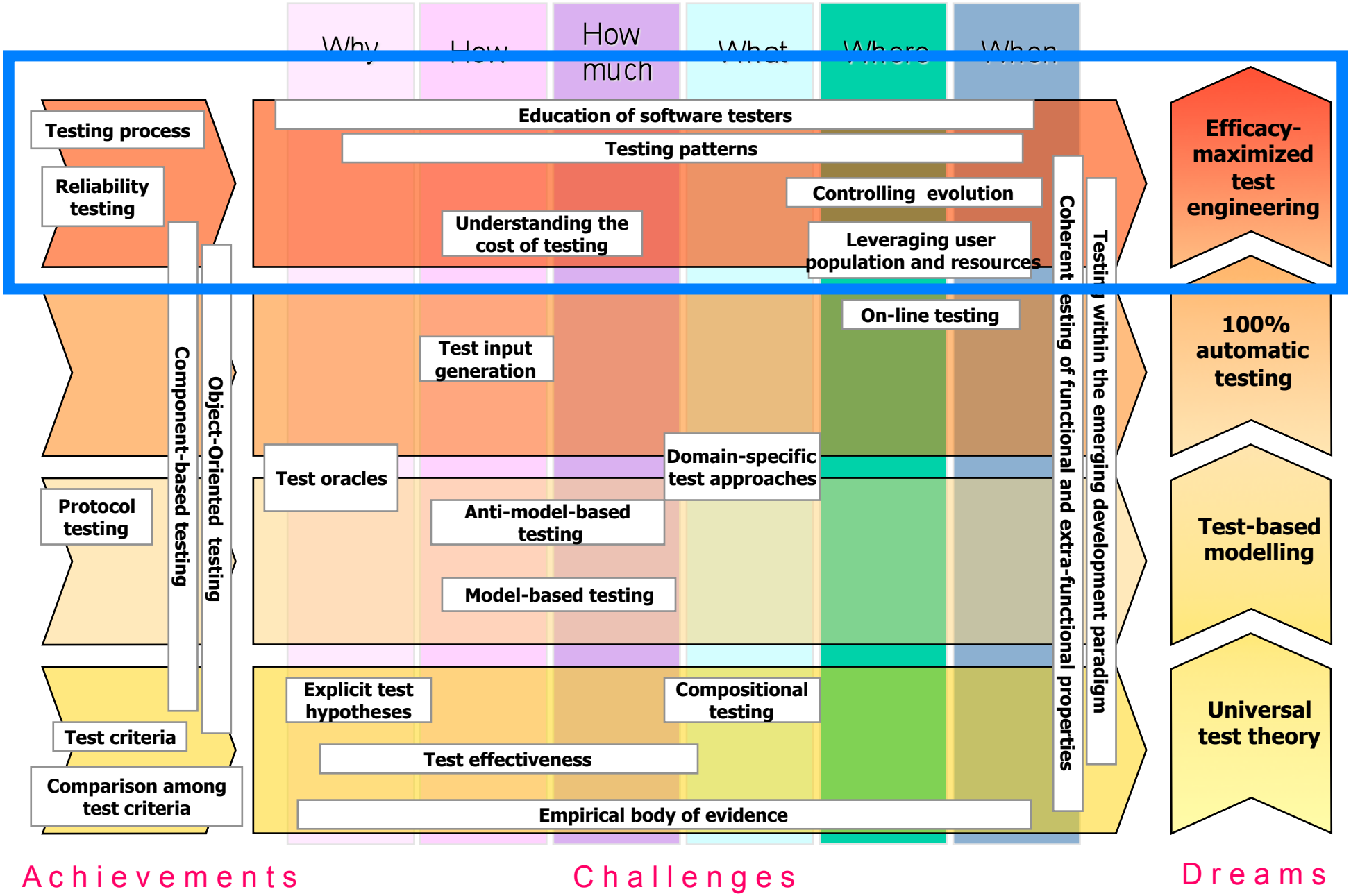


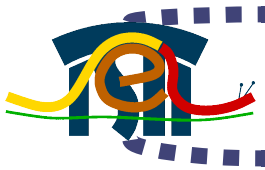
# Challenges to 100% Automatic Testing

- Domain-specific test approaches
  - how to exploit domain knowledge to make testing more effective: specialized modeling and tools (overlaps with test-based modeling)
- Test input data generation
  - results promised by MBT, clever random and search-based techniques (variously combined)***
- On-line testing
  - a.k.a. monitoring or passive testing: when sampling in advance is not feasible
- ...

***Revised in new roadmap

# Software testing research roadmap 2007

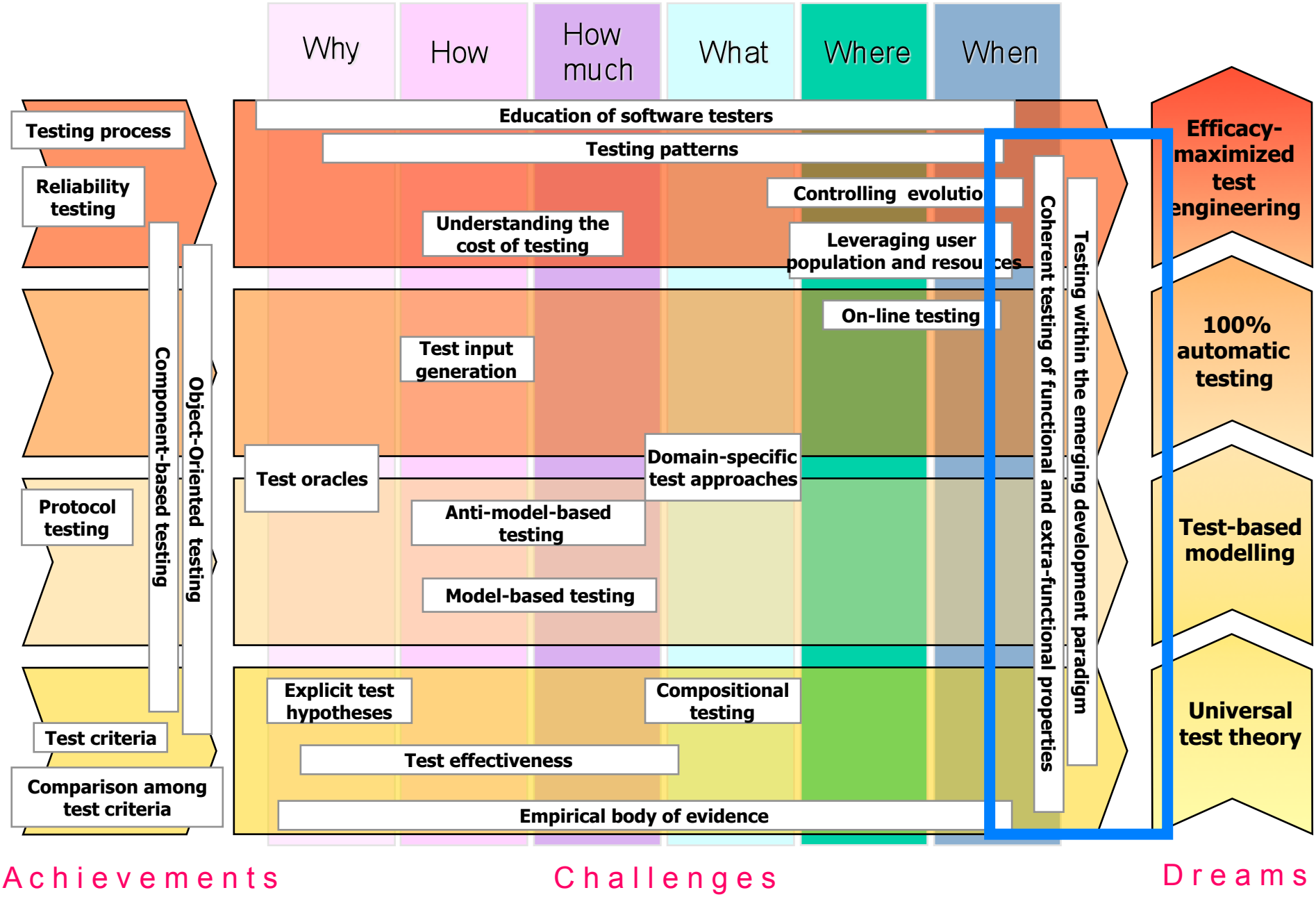


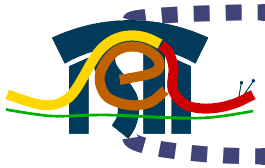


# Challenges to Efficacy-maximized Test Engineering

- Controlling evolution
  - Maximize test re-use, whereby regression testing becomes the norm - orthogonal to compositional testing (also related test factoring: how to convert a long test into small unit tests)
- Leveraging user population and resources
  - collect field data, while coping with performance and security issues
- Testing patterns
  - catalog well-proven solutions to recurring problems
- Understanding the costs of testing
  - as per the recent paradigm of Value-based SE
- Education of software testers
  - Continuous education to novel approaches
- ...

# Software testing research roadmap 2007

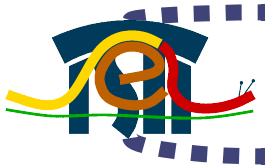




# Transversal Challenges

- Testing within the emerging development paradigm
  - testing technology has to follow/adapt to the currently used technologies, e.g., today the focus is on testing of Service-oriented applications
- Coherent testing of functional and extra-functional properties
  - with increasing software pervasiveness, the quality of service becomes more and more important, we need ways to test extra-functional – harmonized with functional testing
- ...

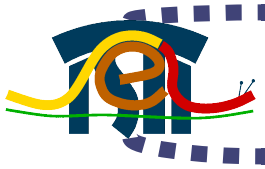
# Re-check/update to 2007's roadmap



# Mapping study

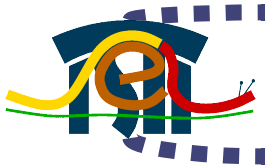
- A mapping study (or scoping review) is a "more 'open' form of SLR [Systematic Literature Review], intended to 'map out' the research that has been undertaken rather than to answer a detailed research question . … Such a study is intended to identify 'gaps' in the set of primary studies, where new or better primary studies are required, as well as 'clusters' where there may be scope for more complete SLRs to be undertaken."

Budgen, D., Turner, M., Brereton, P. & Kitchenham, B. (2008), Using Mapping Studies in Software Engineering, Proc. PPIG 2008, Lancaster University, pp. 195-204.



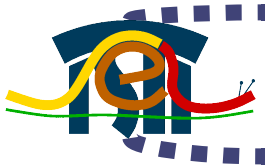
# The “unsystematic” mapping study

1. Study scope
2. Search & Screening
3. Obtain a set of papers
4. Classification & Mapping
5. Analysis/implications



# Study scope

- Wanted to validate and update 2007 roadmap
- Research in ST
- Papers published in the period [2008-2011]
- Setting boundaries:
  - Identify most researched topics against FOSE 2007 roadmap
  - Identify topics that are not represented in FOSE 2007 roadmap
  - Identify topics that are no longer relevant

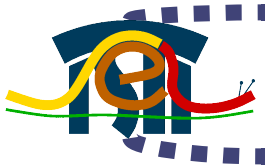


# Search & Screening

- Papers included in **The DBLP Computer Science Bibliography: http://www.dblp.org/**

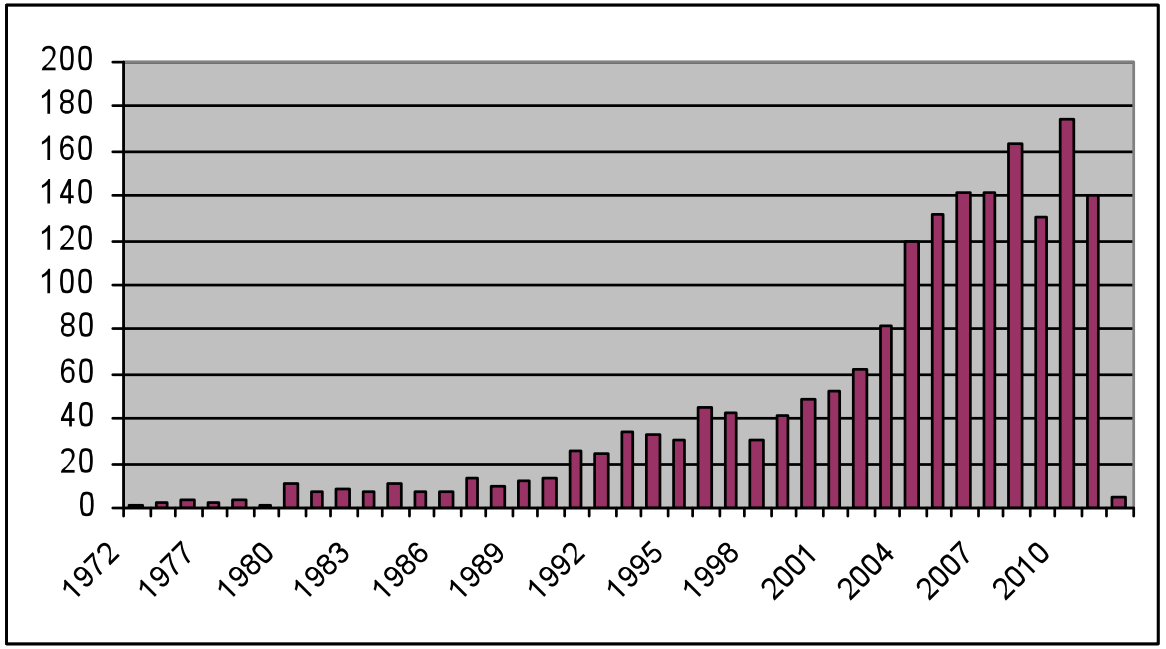
- Search for "software + testing":
  - Warning: very limited subset (e.g. Antonia Bertolino has 19 out of her 98 entries), yet hopefully "representative" (not biased ??)

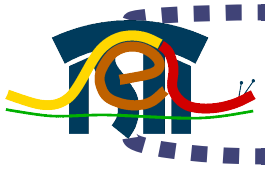
- Restrict by year, select 2008, 2009, 2010, 2011.



# Set of papers

- In total 1823 papers, of which:
  - 164 in 2008
  - 131 in 2009
  - 173 in 2010
  - 139 in 2011



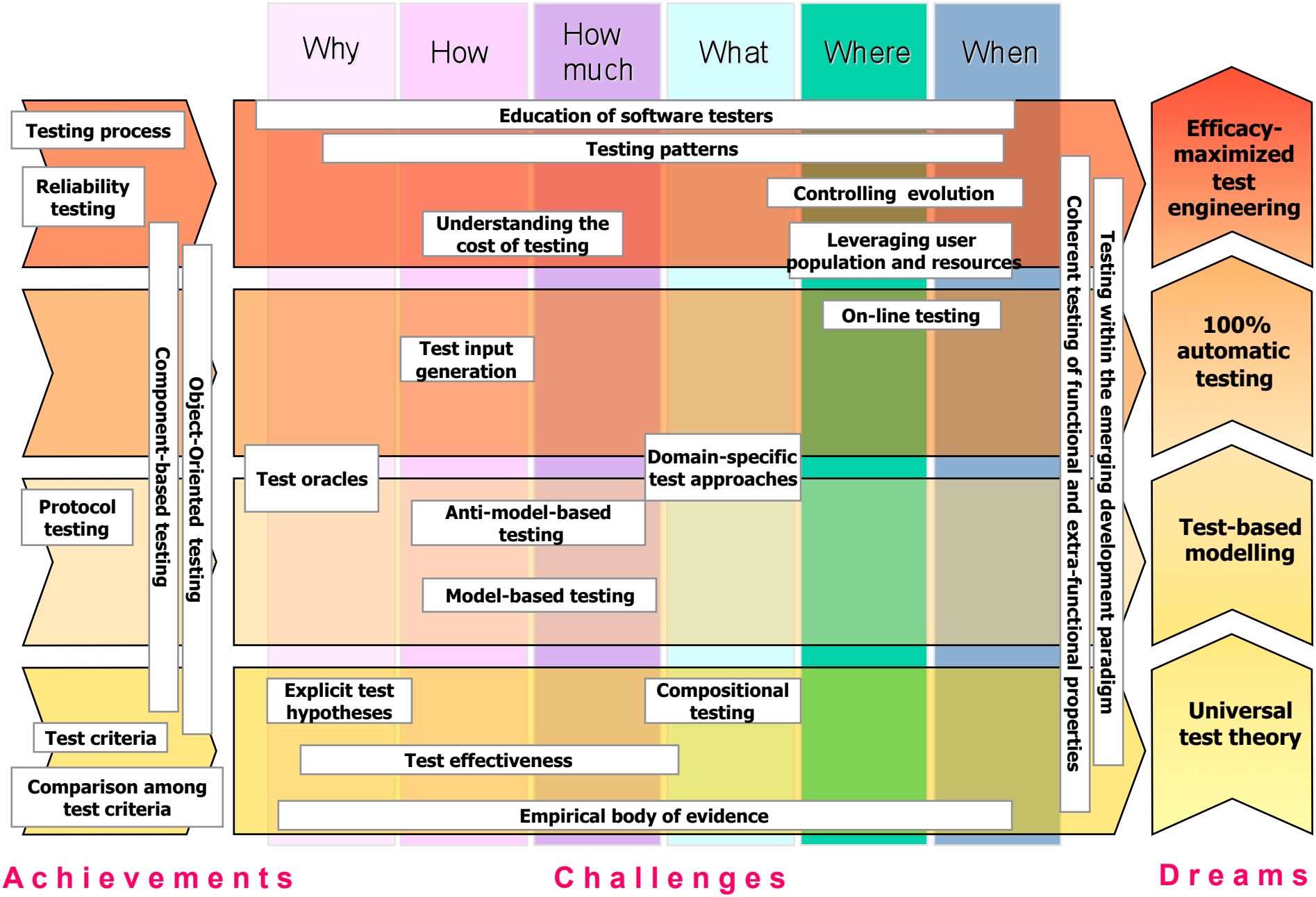


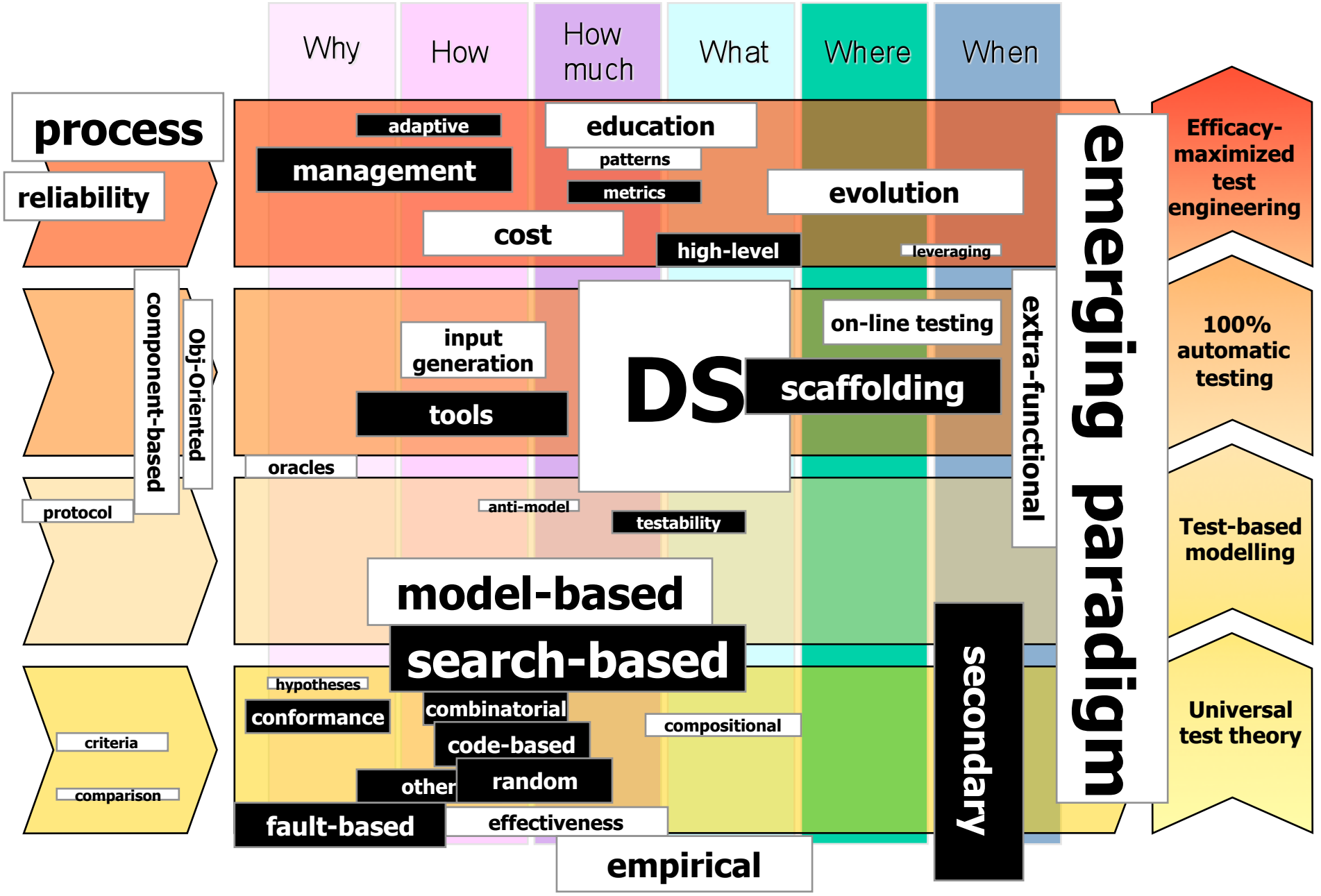
# Classification

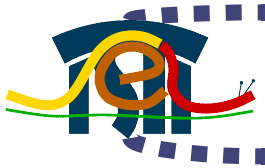
- Categories are those in FOSE 2007, as a start
- Look at paper titles, and if unclear read abstract
- Assign papers to categories
- If no map, create new category
  - Several validity threats:
    - papers can generally go under more than one category
    - If a new category is introduced, no iteration

# Roadmap 2012

# Software testing research roadmap 2007




Why
How
How much
What
Where
When
process
reliability
adaptive
management
education
patterns
metrics
cost
evolution
high-level
leveraging
component-based
Obj-Oriented
protocol
input generation
tools
DS
scaffolding
on-line testing
extra-functional
oracles
anti-model
testability
model-based
search-based
secondary
hypotheses
conformance
combinatorial
compositional
code-based
criteria
comparison
other
random
fault-based
effectiveness
empirical
emerging paradigm
Efficacy-maximized test engineering
100% automatic testing
Test-based modelling
Universal test theory
Achievements
Challenges
Dreams



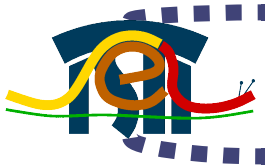
# What was missing in FOSE 2007

Most notably:

- Test techniques:
  - Code-based
  - Combinatorial
  - Random
  - Fault-based
  - Search-based
  - …
- Tooling
- Secondary studies
- …

# Analysis 1

- The study has been informal and driven by research literature: the outcome does not (and could not) answer whether such roadmap reflects well industry and practice needs
  - For example, are more actively researched topics more relevant in practice, or they are just more attractive from a scientific perspective?
- Classification is arbitrary and subject to my own judgment:
  - Some topics would require a finer classification, notably "emerging paradigm" and "DS" (Domain Specific)



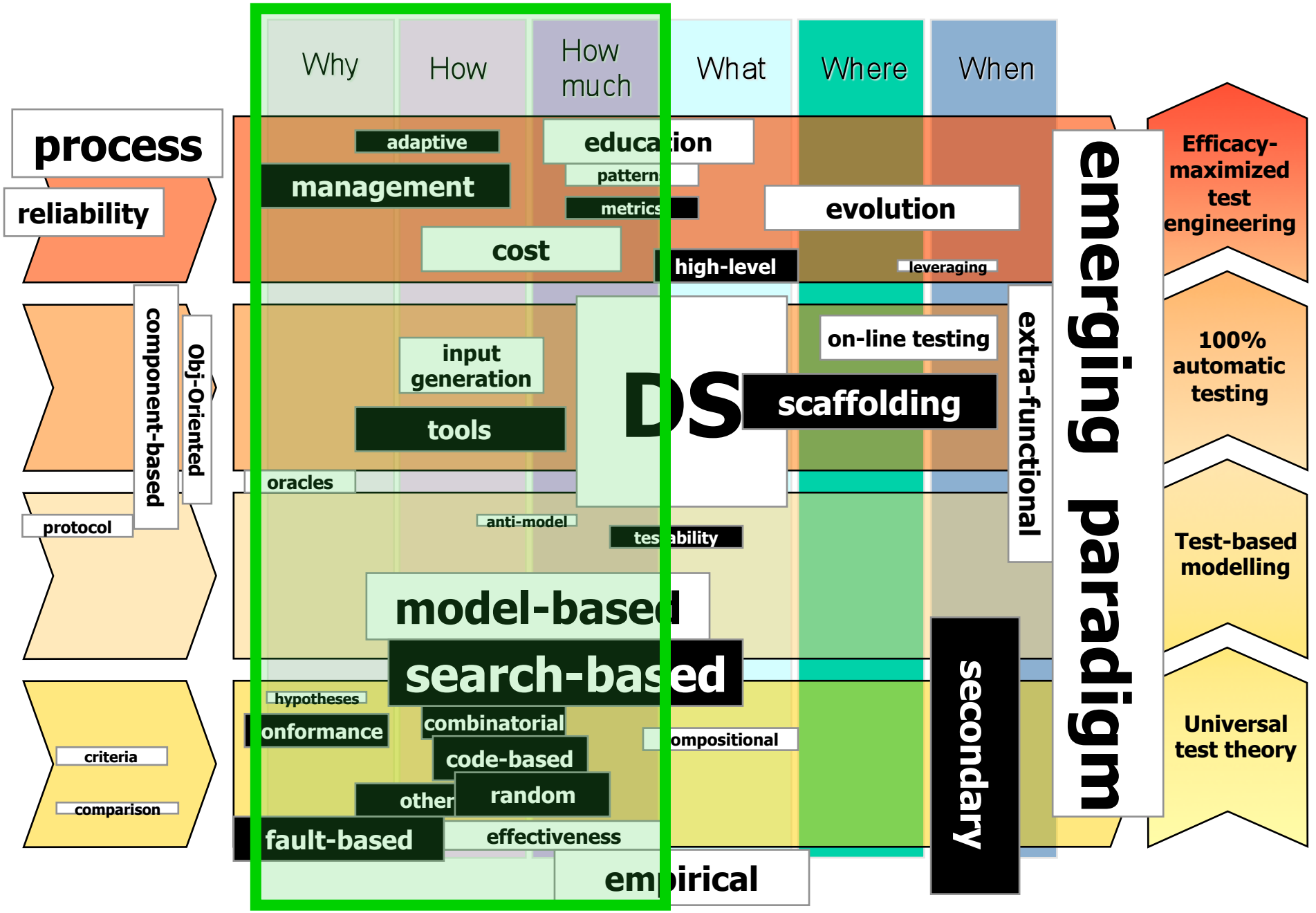
# Analysis 2

- A few research topics that do not (seem to) draw much attention would actually be crucial to advance SOTA towards the dreams

- Notably, among others:
  - Test oracles
  - Test hypotheses
  - Making test compositional

  stay in the relation between testing and verification

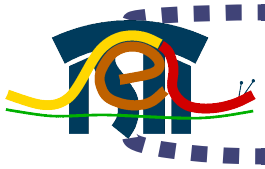
# Testing & Verification


Why
How
How much
What
Where
When
process
reliability
component-based
Obj-Oriented
protocol
criteria
comparison
adaptive
management
cost
pattern
metrics
high-level
leveraging
evolution
input generation
tools
oracles
anti-model
scaffolding
on-line testing
extra-functional
model-based
search-based
hypotheses
combinatorial
code-based
random
other
fault-based
effectiveness
empirical
secondary
emerging paradigm
Efficacy-maximized test engineering
100% automatic testing
Test-based modelling
Universal test theory
Achievements
Challenges
Dreams

# Test theory

J. B. Goodenough, S. L. Gerhart: Toward a Theory of Test Data Selection. IEEE TSE 1(2), 1975: introduce **reliable** and **valid** properties for an ideal testing criterion.

- Concept of (reliable) subdomains and partitions
- Test hypotheses (Gaudel)
  - Verification techniques as a means to formalize test hypotheses
  - Equivalence relation under which the SUT conforms to the reference model



# The tester dilemma

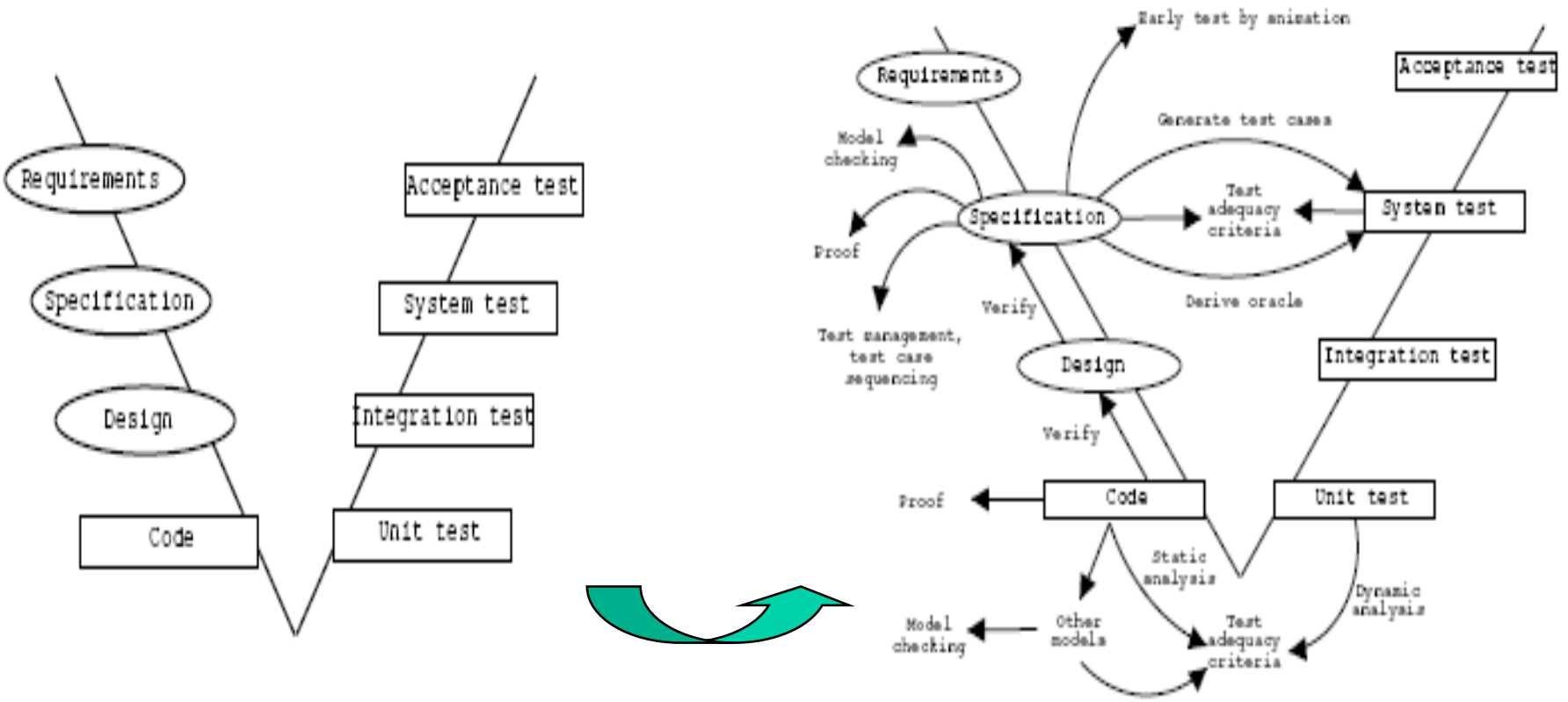
A- The more we test the more bugs we find

OR

B- The more we test the less bugs we find

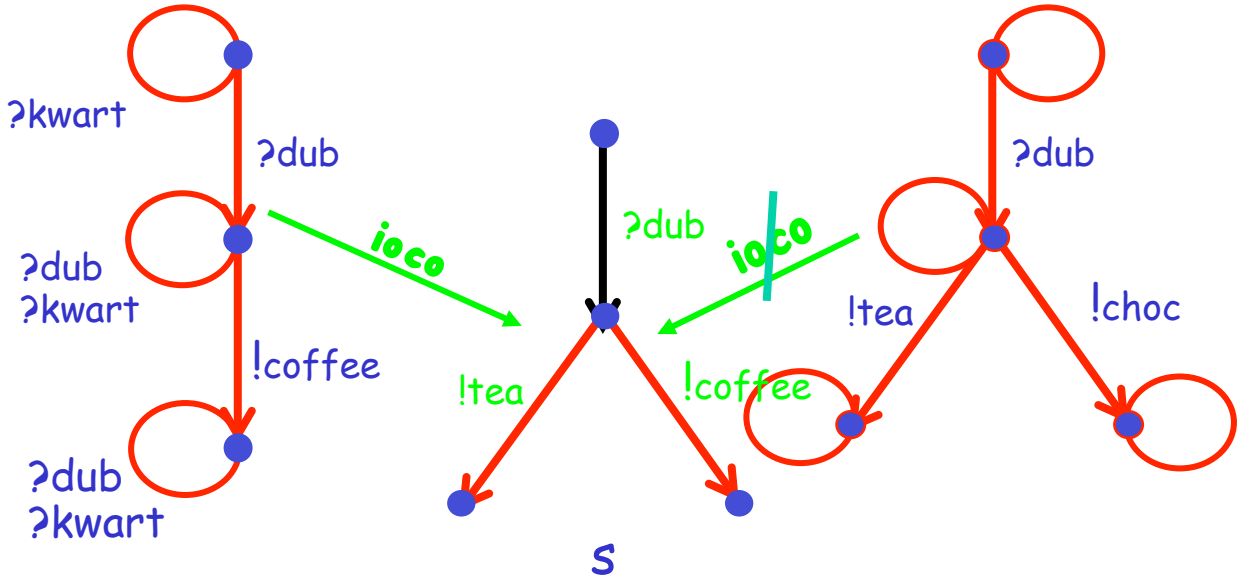
Which one would you prefer?

# Model-based testing




R.M. Hierons, K. Bogdanov, J.P. Bowen, R. Cleaveland, J. Derrick, J. Dick, M. Gheorghe, M. Harman, K. Kapoor, P. Krause, G. Lüttgen, A.J.H. Simons, S. Vilkomir, M.R. Woodward, and H. Zedan. Using formal specifications to support testing. ACM Comput. Surv. 41(2), 2009

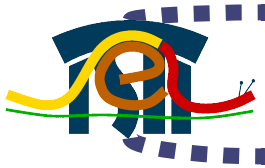
# Test oracles




Example from Jan Tretmans's ioco relation

```
// oracle for ordered set
for (int i = 0; i<size-1; i++)
assert(a[i]<=a[i+1]);
```

Specification-based assertions

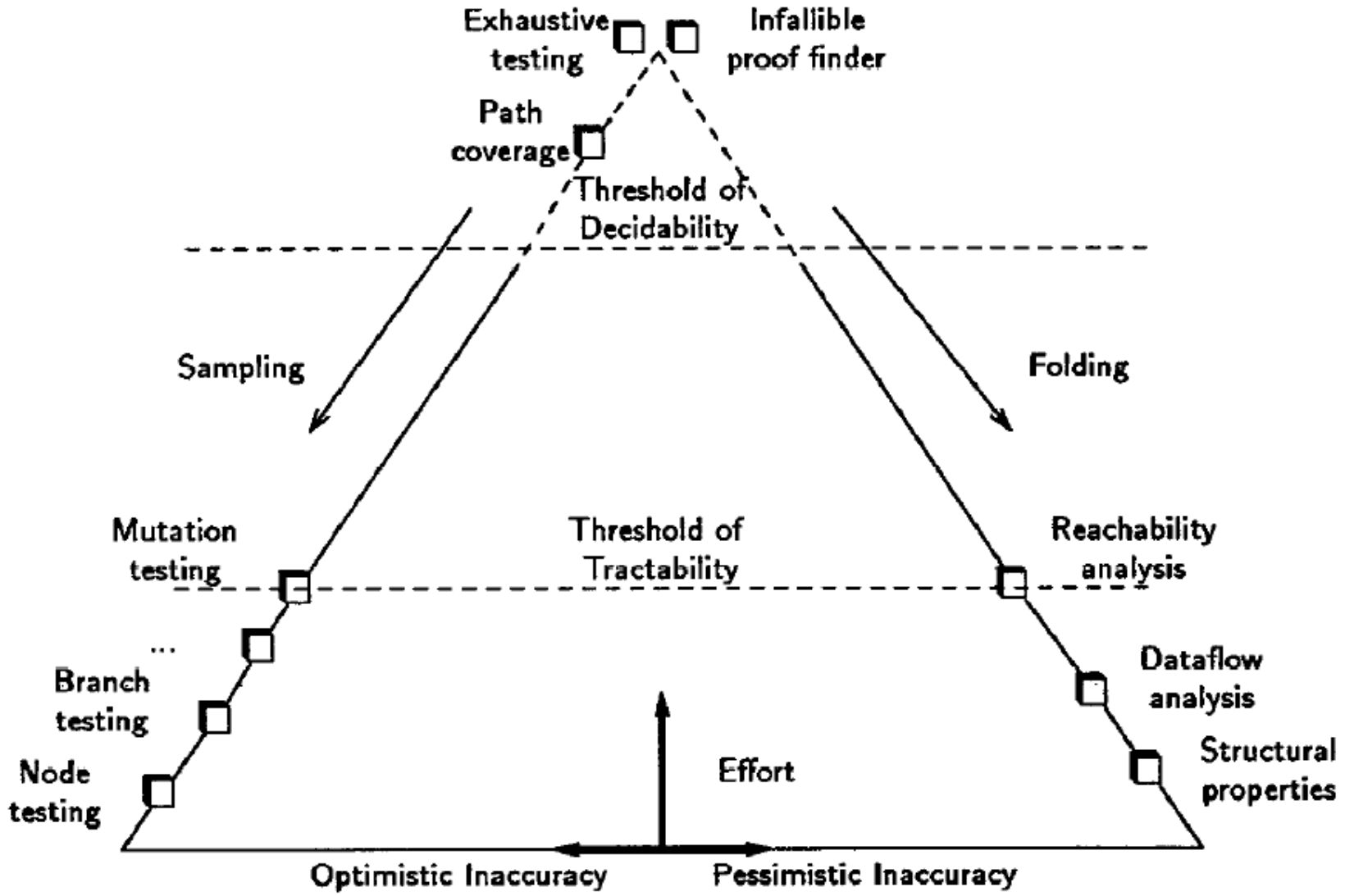


# Combining Testing and Verification

- **"little guidance has been offered on how to combine V&V methods to achieve a synergy that is greater than the sum of the value of applying individual methods "**
  - How can testing be used to discharge assumptions underlying a formal methods verification..
  - How can formal analysis be used to transform a testing problem requiring excessive computational resources (trials) into a testing problem requiring far fewer resources?
- E.g.: a framework postulates the existence of crossover points where an optimal (according to a RGM) V&V strategy would switch from one method to another…

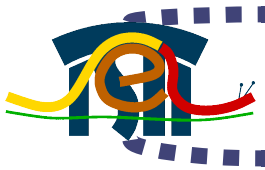
Lowry, M.; Boyd, M.; Kulkami, D., "Towards a theory for integration of mathematical verification and empirical testing," Proc. 13th IEEE International Conference on Automated Software Engineering, 1998.

# Sampling vs. Unfolding




Young M., and Taylor R. 1989. Rethinking the taxonomy of fault detection techniques. In Proc. 11th Int. Conf. Soft. Eng. (ICSE '89). ACM, 53-62.

# Conclusions



# Conclusions

- We have presented the FOSE 2007 roadmap of software testing research; as said: <<the value of the roadmap mostly resides in setting the scene and organizing the concepts>>. Certainly more ST research challenges (current and future) can find their place in the picture.
- We have revisited the roadmap labels, weighting and augmenting them.
- We have attempted to find some interesting venues for a synergy between ST and verification, as a trigger for discussion.
- For future: make "unsystematic" systematic and document process, so to make it more valuable

COW: Software Testing and Verification

# Recent trends in software testing research

an unsystematic (road)mapping study