

31. Different Types of Research Hypotheses, Questions, Methods, and Results in Software Engineering



Prof. Dr. Uwe Aßmann
Softwaretechnologie
Technische Universität
Dresden
2016-0.5, 5/9/16
<http://st.inf.tu-dresden.de/teaching/asics>

- 1) Research Hypotheses
- 2) Different Types of Research Results: Observations, Laws, Theories, Patterns
- 3) Writing Abstracts
- 4) Newman's Template Abstracts

[Library of Congress WPA poster]

Obligatory Literature

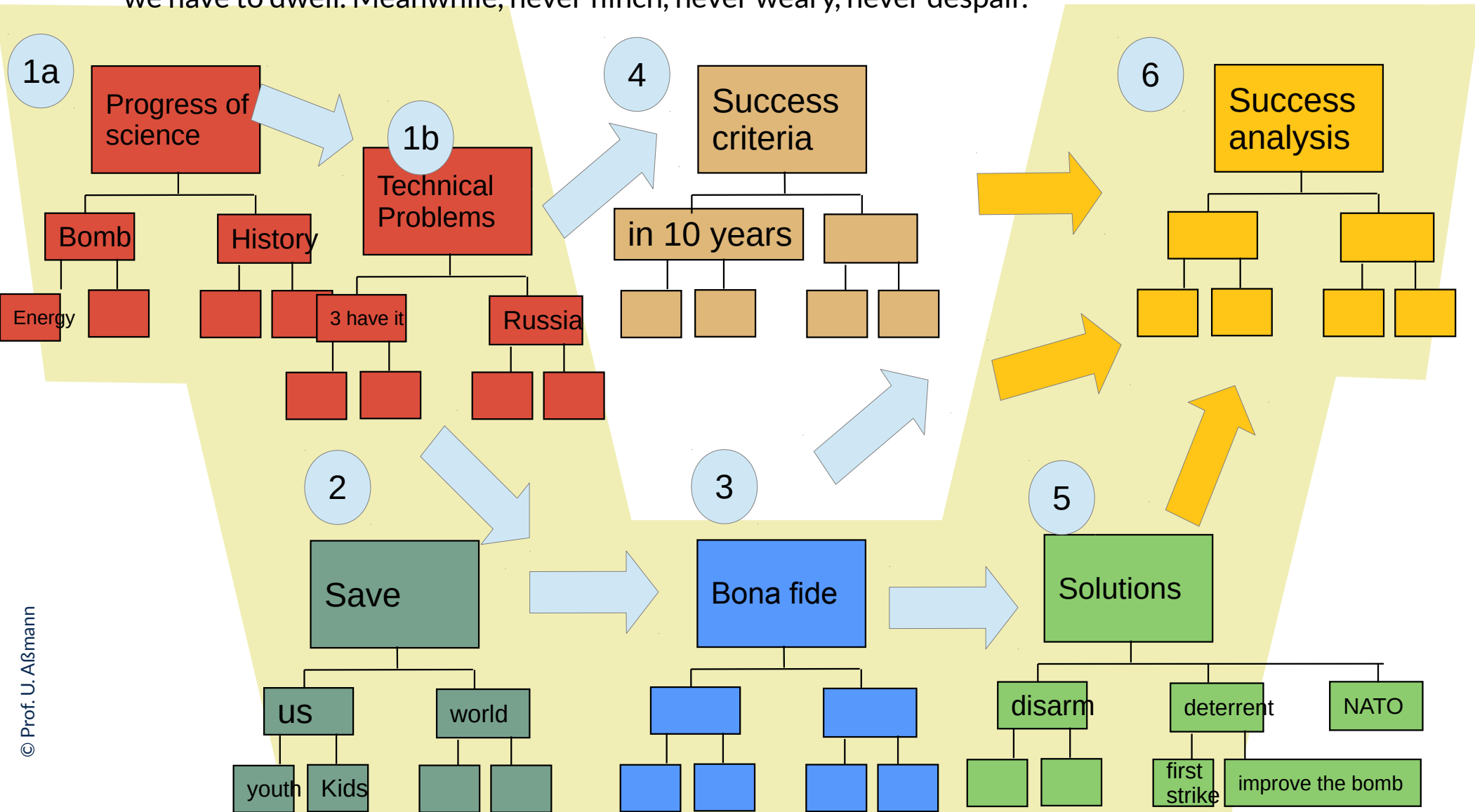
- ▶ [Shaw-Research] Mary Shaw. What makes good research in software engineering? Int. Journal of Software Tools for Technology Transfer (STTT), 4(1):1-7, 2002.
- ▶ [Shaw-ETAPS02] Mary Shaw. Slide set of key note at ETAPS 2002. Good summary of [Shaw-Research]
- ▶ Mary Shaw's web site <http://spoke.compose.cs.cmu.edu/shaweb/>
- ▶ [Bundy] Alan Bundy. How to Write an Informatics Paper. Web page:
 - <http://homepages.inf.ed.ac.uk/bundy/how-tos/writingGuide.html>

References

- ▶ Dieter Rombach. Klaus Endres. A Handbook of Software and Systems Engineering. Addison-Wesley.
- ▶ [Xu-Nygaard] Dianxiang Xu and Kendall E. Nygaard. Threat-driven modeling and verification of secure software using aspect-oriented petri nets. IEEE Trans. Software Eng, 32(4):265-278, 2006.
- ▶ Fun:
 - Scientific Balloons
 - http://www.centennialofflight.gov/essay/Dictionary/Scientific_Balloons/DI72.htm

BATE-POPP on "Never Despair"

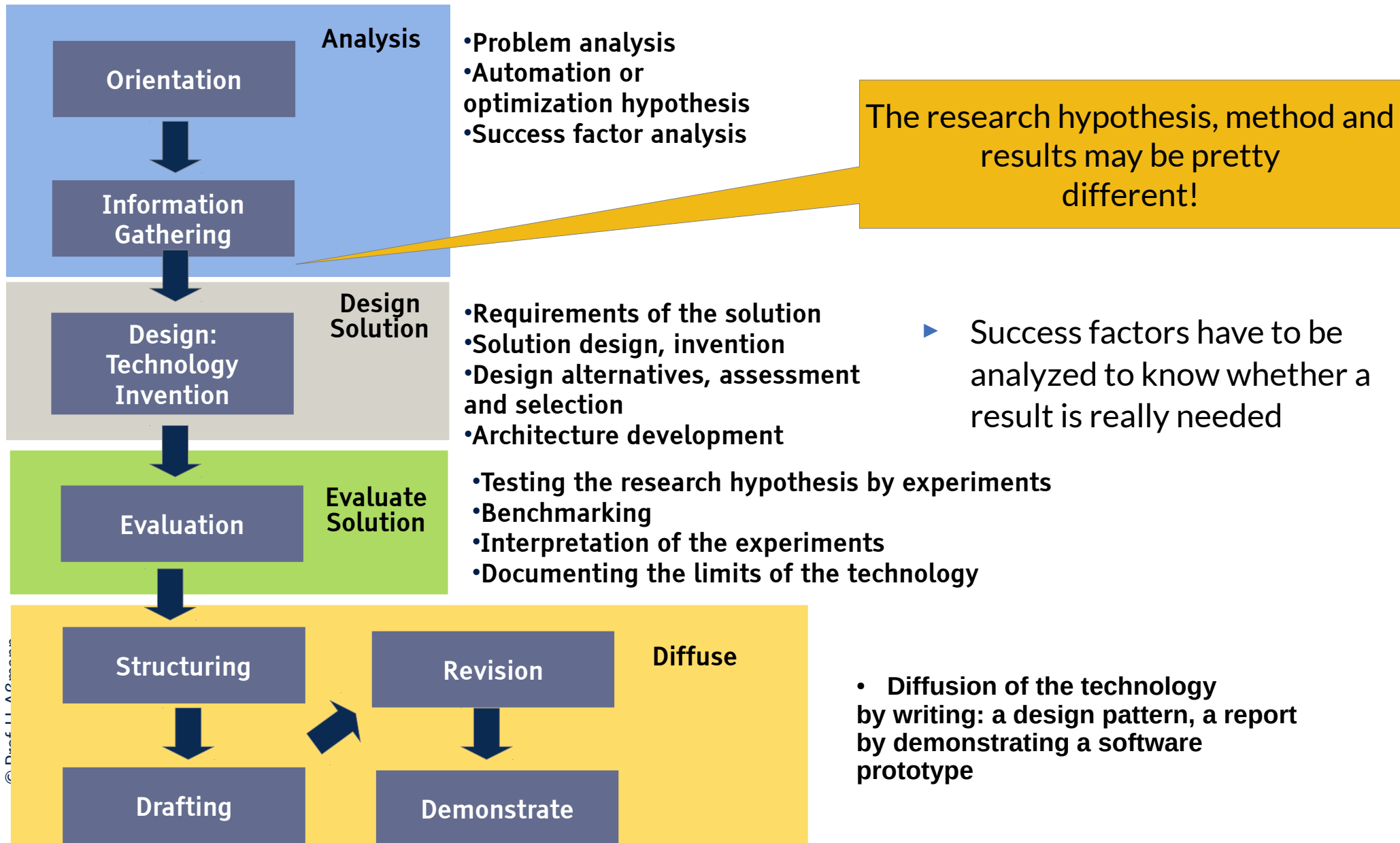
"The day may dawn when fair play, love for one's fellow-men, respect for justice and freedom, will enable tormented generations to march forth serene and triumphant from the hideous epoch in which we have to dwell. Meanwhile, never flinch, never weary, never despair."



31.1 Different Kinds of Research Hypotheses

- ▶ We dive now into the writing of research papers
- ▶ A research paper is very much linked with a research hypothesis (question, success criterion, result)

The ADED Research Process for Technical Science Thesis

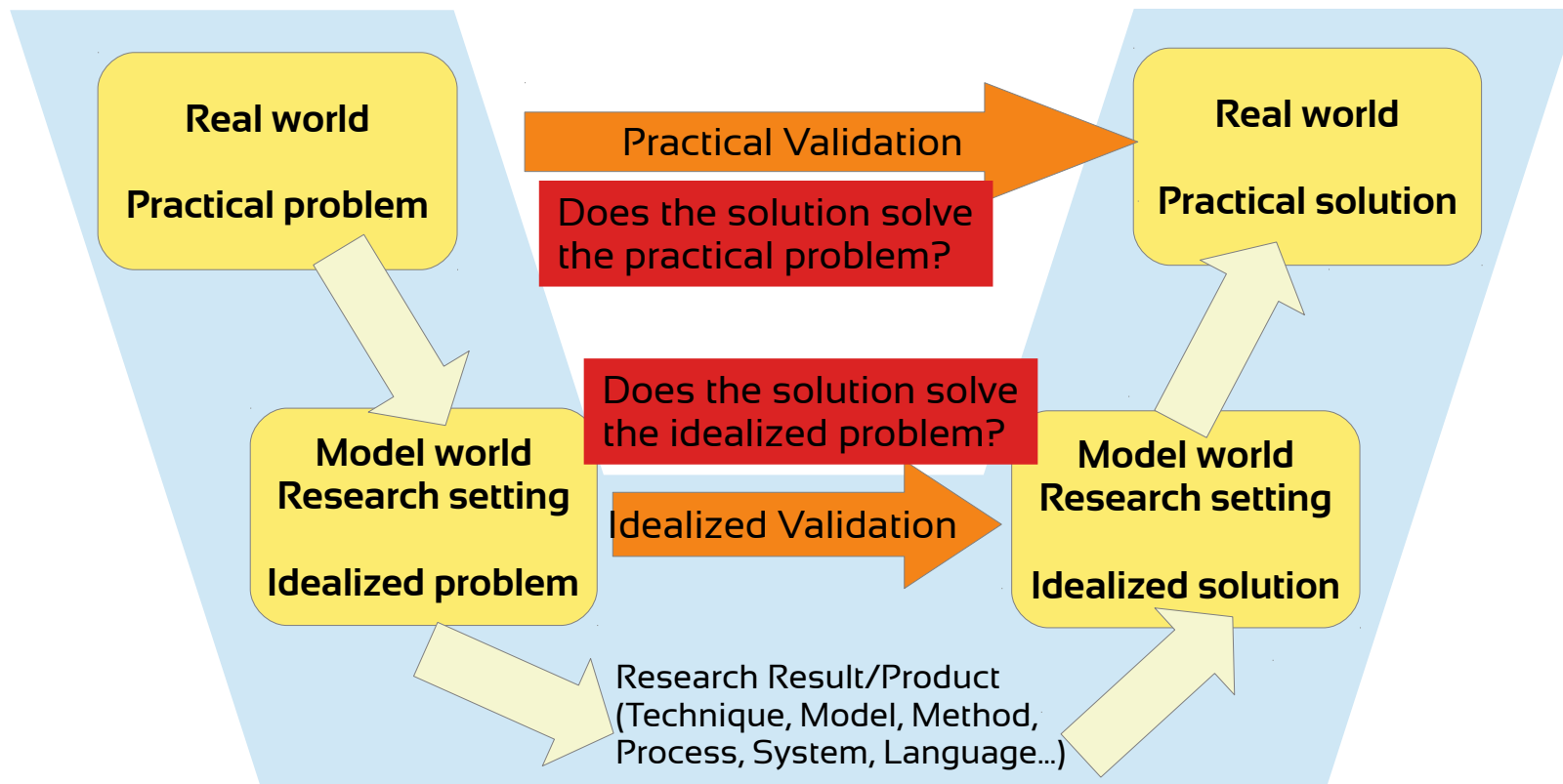


Remember: Practical Research vs. Idealized Research

7

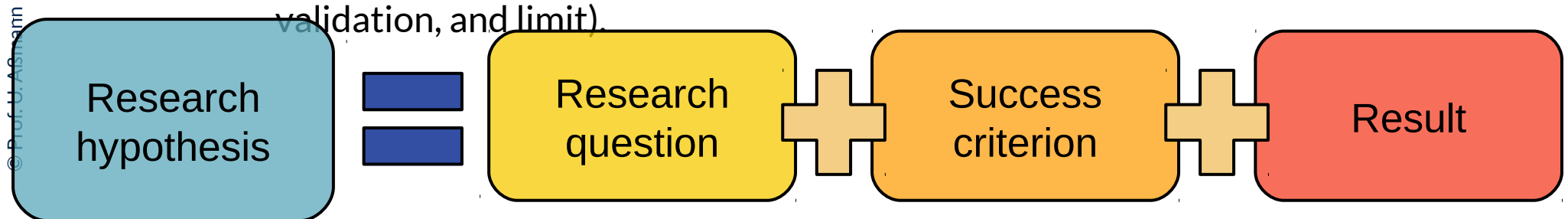
Academic Skills in Computer Science (ASICS)

- ▶ [Shaw-ETAPS02] Many research papers and solutions require a *model of reality* in which their result is valid. A **model of reality** is an idealized abstraction of reality
- ▶ An **idealized research problem** is a research problem in a model of reality, a **complete (practical) research** result solves a practical research problem
- ▶ Structural science (mathematics, theoretical computer science, computer science) works in idealized model worlds
- ▶ Technical science (engineering science), also Software Engineering, works for **practical problems** and must research **practical solutions**
- ▶ **Technical scientists and Engineers** have to produce **practical solutions**



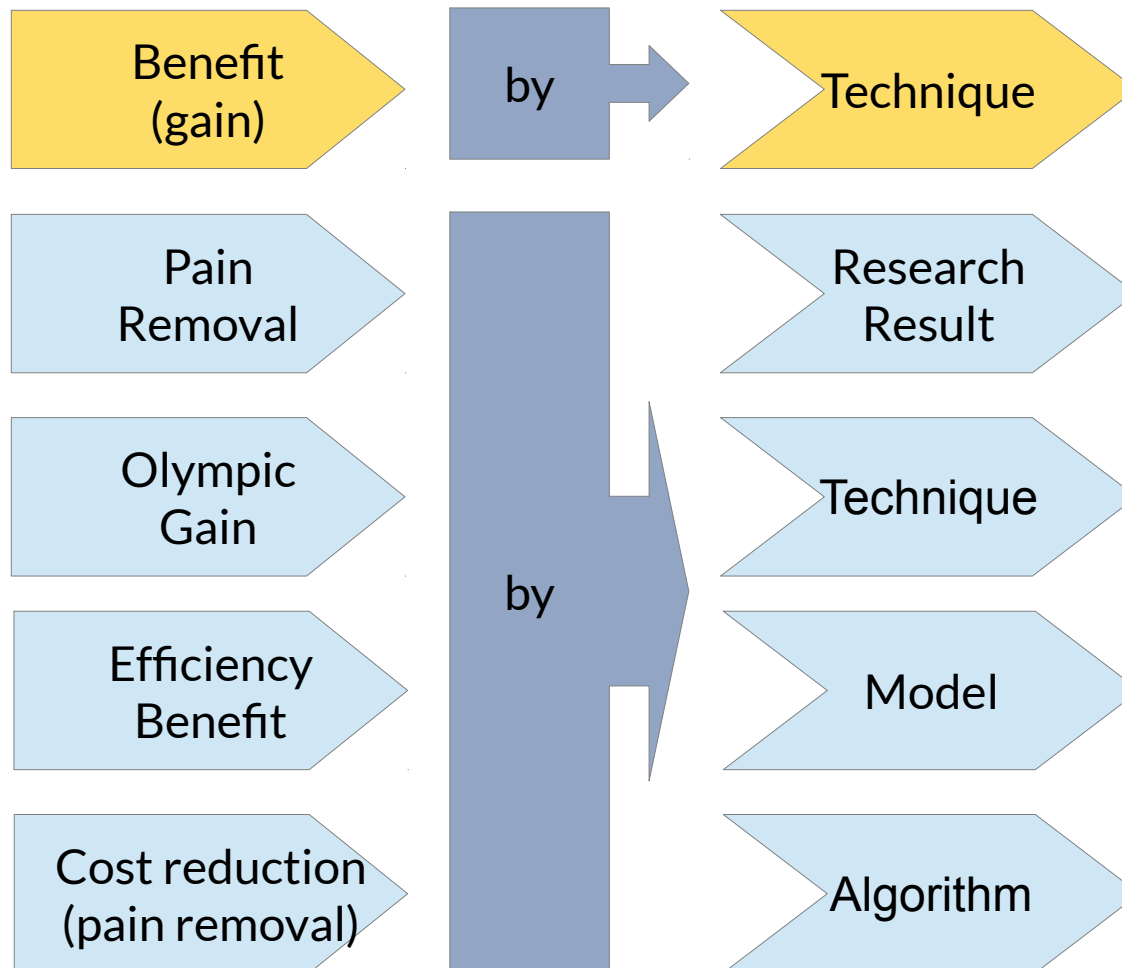
Relationship of Research Hypothesis and Question

- ▶ The relationship of a research question to a research hypothesis is similar to the relationship of a thesis question and a thesis of a text block.
 - Text Question creates interest, thesis answers it. Text Thesis has topic and controlling idea
 - **Research hypothesis** has a *research result* (topic) and a *success criterion, research result*
 - **Controller hypothesis** has additionally a *research method* and *research valuation* (development scheme), and a *limit*
- ▶ Important: Your research hypothesis and research question has to be found *crystally clear* in your **introduction** and your **abstract**
 - Every time, you refine the hypothesis, you have to rewrite the introduction and the abstract
 - Your **slides of the defense** also have to display them *crystally clear*
 - Make **one** slide to present hypothesis (question, success criterion, result, validation, and limit).



A Technical Science Hypothesis has “Benefit by Technique”

- ▶ A **Technical Science Hypothesis (Technik-Hypothese)** is an hypothesis about *achieving a benefit (solving a problem, solving a research problem or reaching an objective) with a technique (technical science research result)*
- ▶ It forms the basis of a technical science paper, Master or PhD thesis



Basic Kinds of Research Hypotheses

- ▶ [Bundy] “The key to successful paper writing is an explicit statement of both a scientific hypothesis and the evidence to support (or refute) it.
- ▶ In experimental research, hypotheses typically take one of the following two forms:
 - 1) Technique/system/solution X **automates** task Y for the first time;
 - 2) Technique/system/solution X automates task Y **better**, along some dimension, than each of its rivals;..”
- ▶ Of course, this holds particularly for software engineering

Automation
hypothesis

Optimization
hypothesis

Important Classes of Research Hypotheses (and Corresponding Success Criteria) in Technical Science

- ▶ How can I automate a technique?
- ▶ What is an engineering technique for a problem?

Constructive Existential
(Automation hypothesis)

- ▶ What is an engineering process for this problem?

- ▶ How can we be faster, go farther, higher?

Farer, Higher,
Wider (Olympic,
absolute)

Optimizing (enhancing, better)
(experimentally better)

- ▶ How can we be more efficient (better cost-utility function)?

Efficiency
(Quality: Utility
vs Resources)

Bundy's Dimensions of Enhancement (Optimization)

The dimensions of enhancement (optimization) are typically [Bundy] and they cover different forms of requirements on a software/technique/solution:

Functional requirements

- ▶ **Behaviour (fulfilling the success criteria):** Solution X has a higher success rate than solution Y, X meets the success criteria better, X has better metrics, X passes the success threshold
 - **Olympic success criteria:** solution gives more utility or has less costs
 - **Utility:** X is faster, X is more precise (BeNiSiLo dimensions)
 - X is shorter, is easier to understand, is easier to write, is more similar to human outputs...:
 - X produces better quality outputs than Y
 - **Cost:** X uses less space, energy, time, resources than Y
 - **Efficiency success criteria:** X is giving more utility *with* less cost (CoTiQQ dimensions)
- ▶ **Coverage (of functional requirements and success criteria):** X is applicable to a wider range of examples than Y. X fulfills more success criteria than Y.

Semi-functional requirements

- ▶ **Dependability:** X is more reliable, safe or secure than each of its competitors

Non-functional requirements

- ▶ **Developer NFR: Maintainability:** Developers find X easier to adapt and extend than its alternatives.
- ▶ **User NFR:**
 - **Useability:** Users find X easier to use than Y.
- ▶ **Management NFR:**
 - **ROI:** solution X achieves “return of investment” earlier than Y

Important Classes of Research Hypotheses (and Corresponding Success Criteria) in Technical Science

- ▶ How much better is this method/technique in (industrial or daily) practice? (according to usability criteria)

Empirically better

- ▶ Which classes of users, companies can benefit?

- ▶ Does feature F hold?

Existential

- ▶ Can we predict P?

Important Classes of Research Hypotheses (and Corresponding Success Criteria) in Technical Science

- ▶ Where does method/technique M fail?
- ▶ Under which conditions does it not work?
- ▶ Assumptions for result

Limit

- ▶ Where is a gap?
- ▶ Where is an open research question?

- ▶ I discovered the following problems with a well-established scientific method.

Limit removal
(can apply to all others)

- ▶ I show how to remove them.

Different Kinds of Research Hypotheses in a Technical Science (Summary)

▶ **Existential hypothesis:**
something exists.

▶ **Automation hypothesis:**
something can be automated the first time [Bundy]

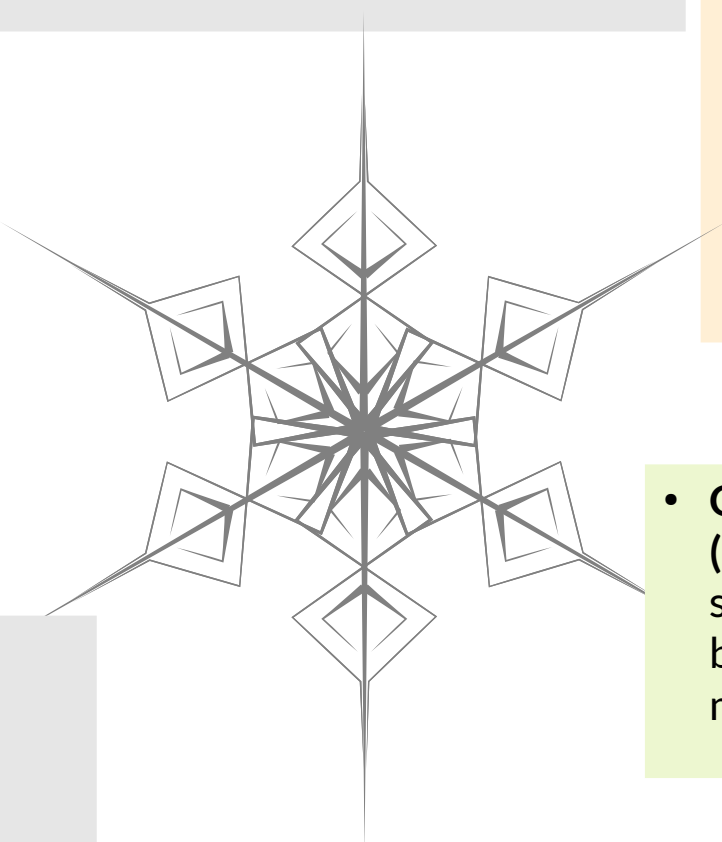
• **Limit removal hypothesis:** my research removes the limits of another method

• **Limit hypothesis:** some other result has its limits

• **Optimizing hypothesis (Enhancement hypothesis):** something can be automated in a better way than with other methods [Bundy]

Olympic hypothesis

Efficiency hypothesis



Different Kinds of Research Hypotheses in a Technical Science (Summary)

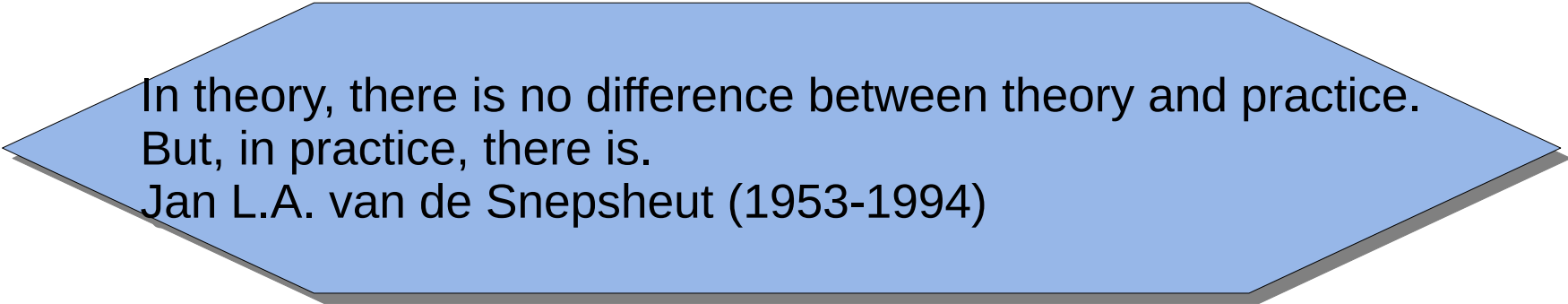
The *benefit* of a technical science hypothesis may be reached in different ways.

- ▶ **Existential hypothesis:** something exists.
- ▶ **Automation hypothesis:** something can be automated the first time [Bundy]
 - Then, you have to show that
 - It is assumed that automation helps
- ▶ **Optimizing hypothesis (Enhancement hypothesis):** something can be automated in a better way than with other methods [Bundy]
 - **Olympic hypothesis:** something can be done faster, wider, higher
 - **Efficiency hypothesis:** something can be done faster, wider, higher with less cost and resource consumption (cost – utility function or relation, economic hypothesis)
 - **Comparison hypothesis:** something A is better than something B. Comparison can be olympic or economic (efficiency-based)
- ▶ **Limit hypothesis:** some other result has its limits
- ▶ **Limit removal hypothesis:** my research removes the limits of another method

Important Classes of Research Questions in Engineering

Type of question/criterion	Examples
Existence?	Does X exist? Does X hold under Y? What is property X of artifact/method Y?
Documenting	What is the current state of X / practice of Y?
Automatable?	What is an automatic way to do/create X? How can we do/create (or automate doing) X?
Olympic? (Quantitative)	How can run X faster? How does X use less memory? How does X spend less energy? How can deliver X more utility? What is a clearer, simpler, more structured design or implementation for application X?
Efficient? (Quality, economics)	How do cost and utility of X relate? How can I increase utility while freezing cost? (better utility) How can I achieve utility while sinking cost?
Comparison	How does X compare to Y?
Limiting	Where does result X not hold? Where is solution X deficient? What are the assumptions under which X holds? What is an open question?



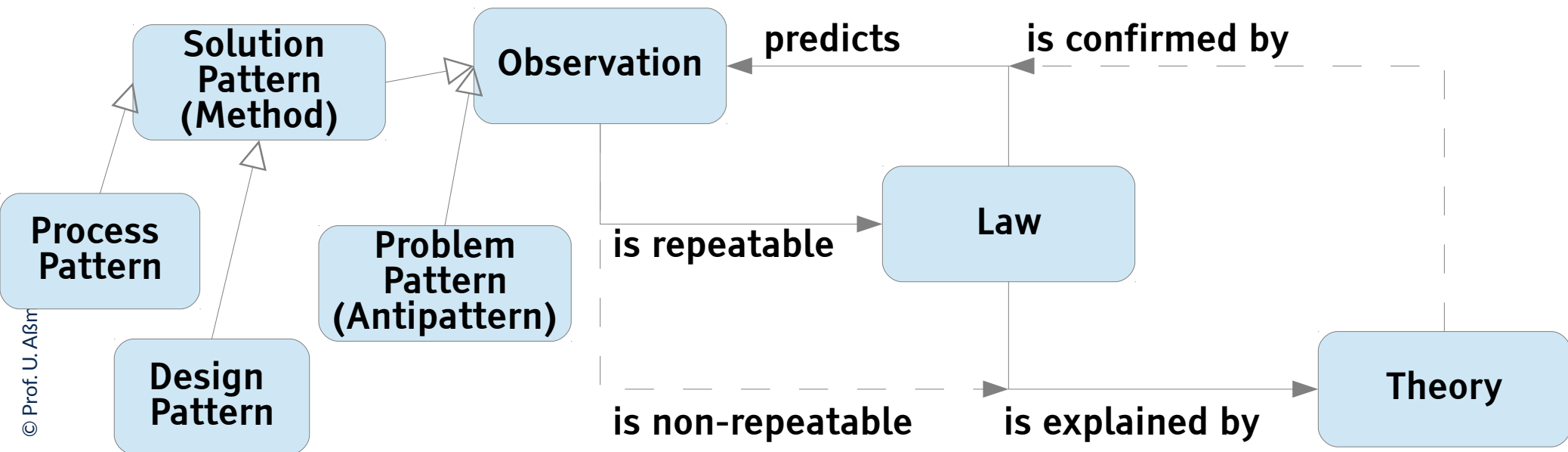


In theory, there is no difference between theory and practice.
But, in practice, there is.
Jan L.A. van de Snepsheut (1953-1994)

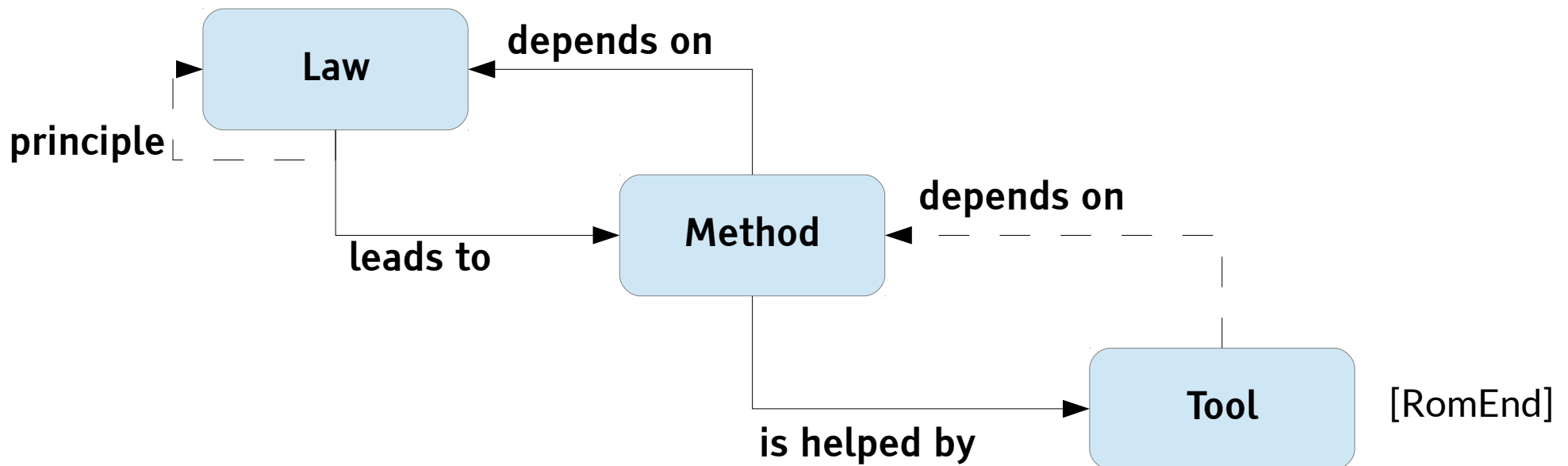
31.2. Different Kinds of Research Results: Observations, Laws, Theories, Patterns

Different Kind of Research Results

- ▶ [RomEnd] collects many research results in software engineering since the 60s. The book suggests also a division of research results (mainly descriptive, analytical and empirical models) into *observations, laws, and theories*.
- ▶ A **law** must lead to the same observation, over and over again.
- ▶ A law does not explain why an observation can be made, instead, a **theory** should explain a law.
 - Theories can be improved over time (see falsificationalism).
 - A theory can consist of a descriptive, analytical or empirical model.
- A **solution pattern** is an observation how to solve a standard problem in a good way



- ▶ A **law** is a claim that leads to repeatable observations, and hence, leads to firm and objective knowledge.
- ▶ A **hypothesis** is a proposition that is tentatively accepted.
- ▶ A **conjecture** is a guess.
- ▶ A **principle** is a basic concept of designing, development, engineering
- ▶ **Techniques** are technical ways to support the work of the software engineer.
- ▶ Processes (procedures) behavioral instructions for the work of the software engineer.
- ▶ **(Best) Practices** are behavioral recommendations to support the work of the software engineer.
- ▶ **Methods (solution pattern)** are procedures, techniques or practices.



31.3 Writing Abstracts and Summaries

Abstracts in structural or technical science should unfold the research hypothesis (question, success criterion, result), the approach and the validation

Abstracts in structural or technical science are most often written with a problem-solving development scheme

- ▶ Abstracts are most often written with a problem-solving development scheme
 - B-POPP, MOPARC, PIBA, ZOPP, BPOPP, NABC, or GulCaramel
- ▶ Abstracts are similar to “elevator pitches”
- ▶ Abstracts can be use all forms of paragraph development scheme
 - direct, smooth direct, or suspended paragraphs
 - Abstracts should not be naked (without point), because then the reader does not get an insight of what the central point of the paper is

Goal: to tell to a potential reader, in the shortest possible space, what he/she will find in the paper. [Gonzalez]

31.3.1 Paper Patterns: Template Abstracts of Newman

- ▶ An abstract should summarize the research statement well.

Patterns of Research Papers

- ▶ A template abstract specializes the “technical science hypothesis” in a specific way.
- ▶ Newman explored them with 5 schemes for the field of Human-Computer Interaction (HCI), but they can be generalized to all disciplines.

A template abstract (pro forma abstract, abstract pattern, paper pattern) is a semantic development scheme for an abstract containing several template sentences. [Newman]

The Template Abstracts of Newman for Classes of Research Papers: Enhanced Model

- ▶ Several template abstracts of Newman suggest olympic or efficiency success criteria.

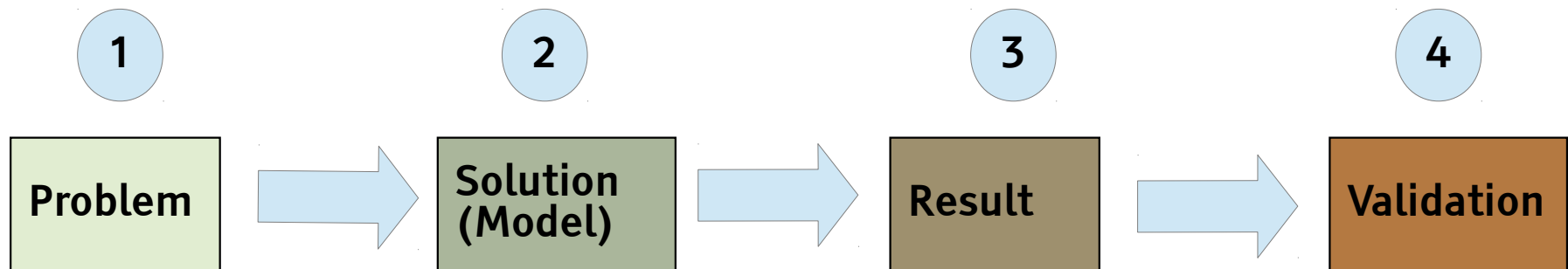
Enhanced Model (EM) (Generalized model, ZOPP for Models):

Problem: Existing <model-type> models are deficient in dealing with <properties> of <solution strategy>.

Solution (Model): An enhanced <model-type> is described,

Result: capable of providing more accurate analyses / predictions of <properties> in <solution strategy> designs.

Validation: The model has been tested by comparing analyses / predictions with empirically measured values of <properties>.



Newman's Enhanced Solution

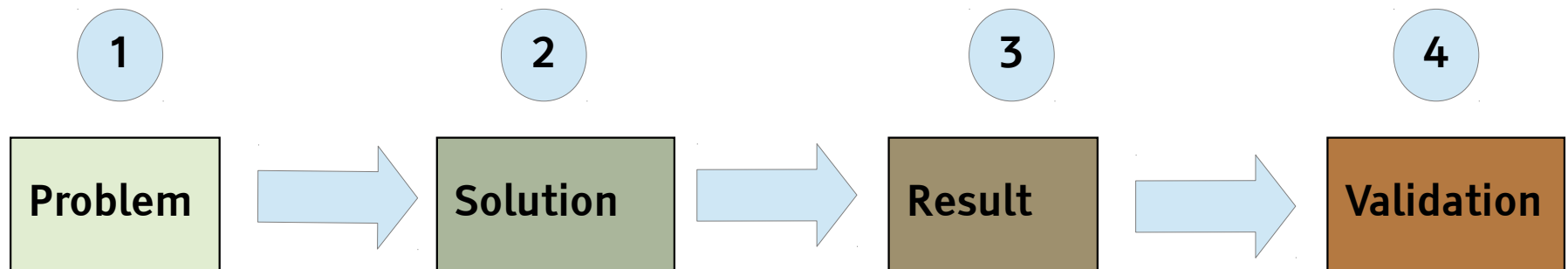
Enhanced Solution (ES): (Better: more olympic or efficient, ZOPP-like)

Problem: Studies of existing <artefact-type> have shown deficiencies in <property>.

Solution: An enhanced design for an <artefact-type> is described, based on <solution strategy>.

Result: In comparison with existing solutions, it offers enhanced levels of <property>, according to analyses based on <model-type>.

Validation: These improvements have been confirmed / demonstrated in tests of a working <artefact-type> based on the design.



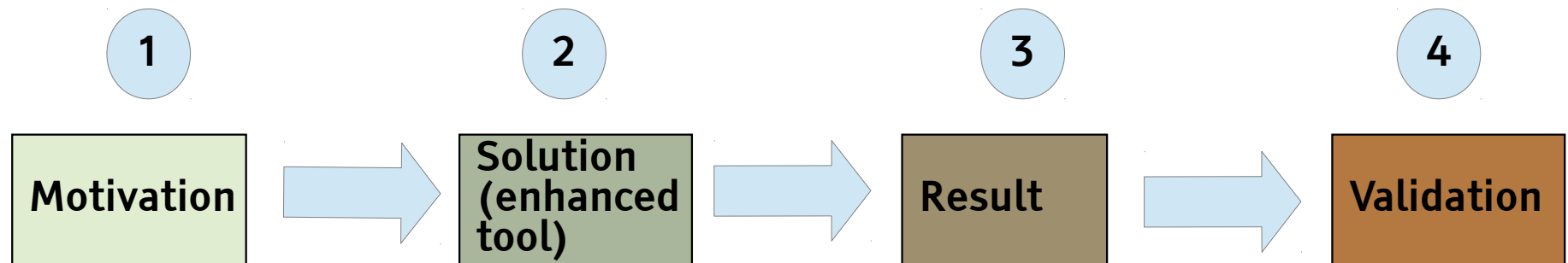
Newman's Enhanced Tool

Enhanced Tool (ET): (Better: more olympic or efficient, MOPARC-like)

Motivation: The effectiveness of <model-type> / <solution strategy> in supporting the design of <artefact-type> has been demonstrated.

Solution (Enhanced tool) and Result: An enhanced tool / method is described for the design of <artefact-type> based on <model-type> / <solution strategy>.

Validation: Examples are provided confirming the effectiveness of its support for <model-type> / <solution strategy> in design.

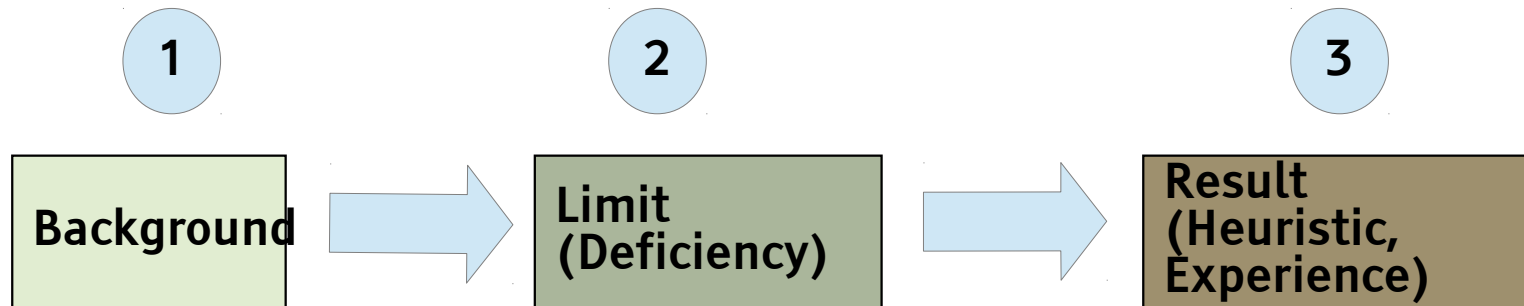


Newman's Experience&Heuristic

Experience and/or Heuristic (XH):

Background: Studies reported here of <application> supported by <supporting technology> generate a number of findings concerning <issues>, including <list-of-findings>.

Limit/Deficiency/Problem: They indicate that <requirement> is / is not met by <design-heuristic>.



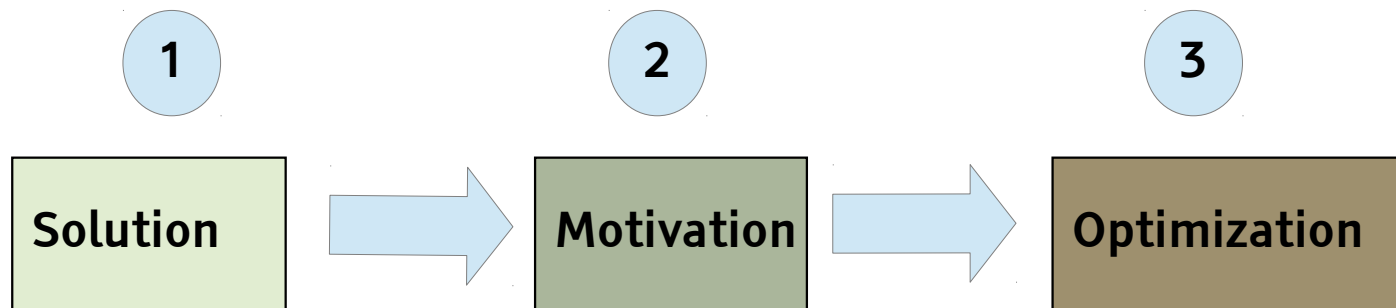
Newman's Radical Solution

Radical Solution (RS, MOP):

Solution: A radical solution to the problem of <problem definition> is described, based on <solution strategy>.

Motivation: In comparison with <existing normal solutions> it offers <advantages>, which have been demonstrated in preliminary tests, but it leaves a number of side-effects to be addressed including <list of side-effects>.

Optimization: Strategies are suggested for addressing these side-effects.



POPP Template Abstracts are Similar to Enhanced Solution

- ▶ Newman's Template Abstracts are nice, but since they are mined from real papers, they have gaps.
- ▶ For instance, instead of EnhancedSolution, we can use B-POPP to have a stronger impression on the reader

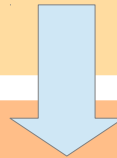
Enhanced Solution (ES): (Better: more olympic or efficient, ZOPP-like)

Problem: Studies of existing <artefact-type> have shown deficiencies in <property>.

Solution: An enhanced design for an <artefact-type> is described, based on <solution strategy>.

Result: In comparison with existing solutions, it offers enhanced levels of <property>, according to analyses based on <model-type>.

Validation: These improvements have been confirmed / demonstrated in tests of a working <artefact-type> based on the design.



B-POPP:

Problem: Studies of existing <artefact-type> have shown deficiencies in <property>.

Goal: <stakeholder group> needs the following <olympic improvements | efficiency improvements | automation>

Blocking factor: So far, the <limit> of <artefact-type> could not be removed.

Success criterion: If <stakeholder-group> can get 20 % of improvement, it will be satisfied.

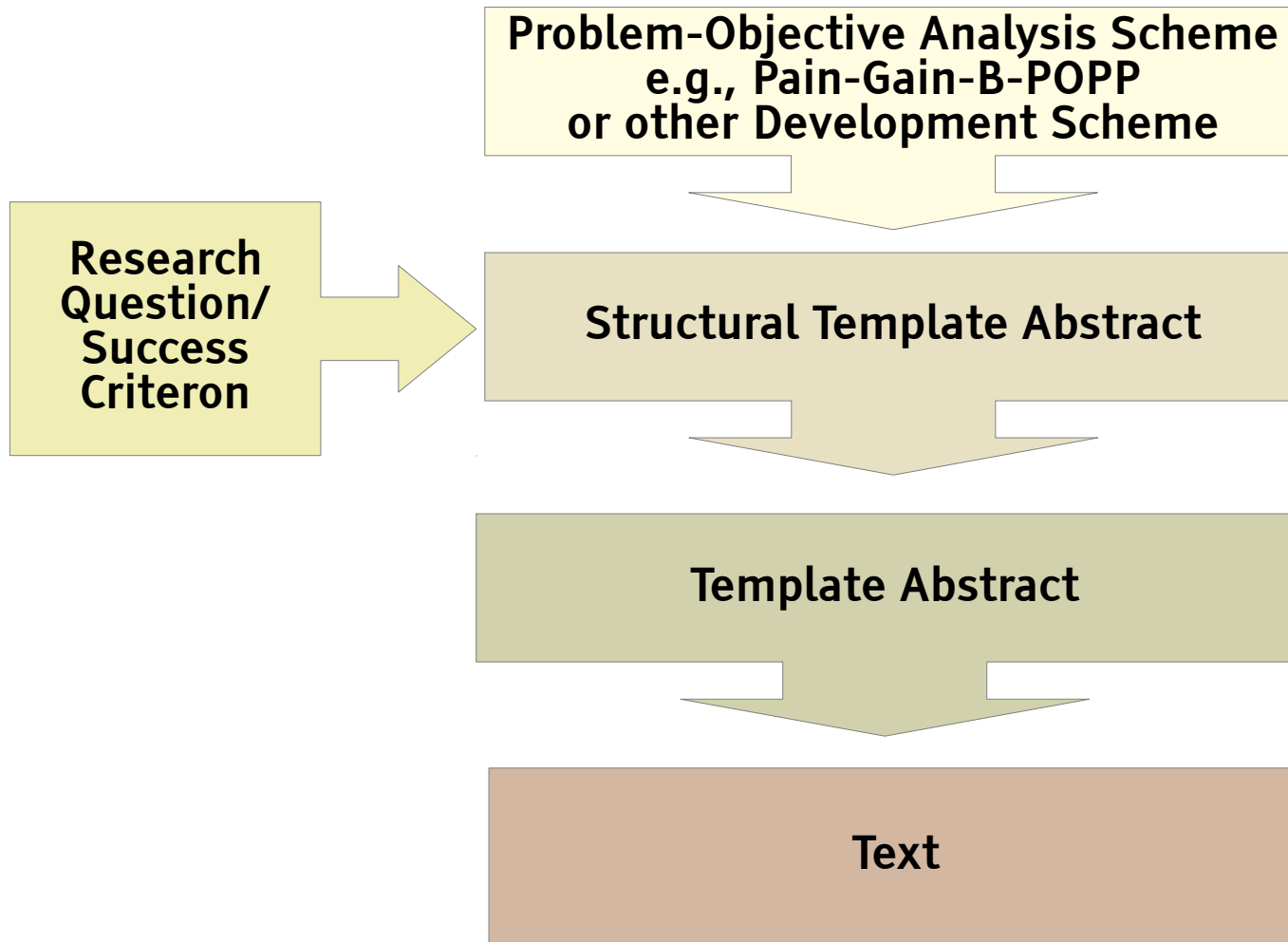
Solution: An enhanced design for an <artefact-type> is described, based on <solution strategy>.

Result: In comparison with existing solutions, it offers enhanced levels of <property>, according to analyses based on <model-type>.

Validation: These improvements have been confirmed / demonstrated in tests of a working <artefact-type> based on the design.

Generation of Template Abstracts: Combination of POA and Template Abstracts

- ▶ “Enhanced Solution” Template Abstracts can be produced by any problem-goal-analysis scheme, or any development scheme, if combined with a olympic or efficiency research question and success criterion.



31.3.2 Further Abstract Patterns

Remember: Newman's Template Abstracts

[Shaw-ETAPS, Newman]

EM: Enhanced model

Existing model-type models are deficient in dealing with properties of solution strategy. An enhanced model-type is described, capable of providing more accurate analyses / predictions of properties in solution strategy designs. The model has been tested by comparing analyses / predictions with empirically measured values of properties.

ES: Enhanced solution

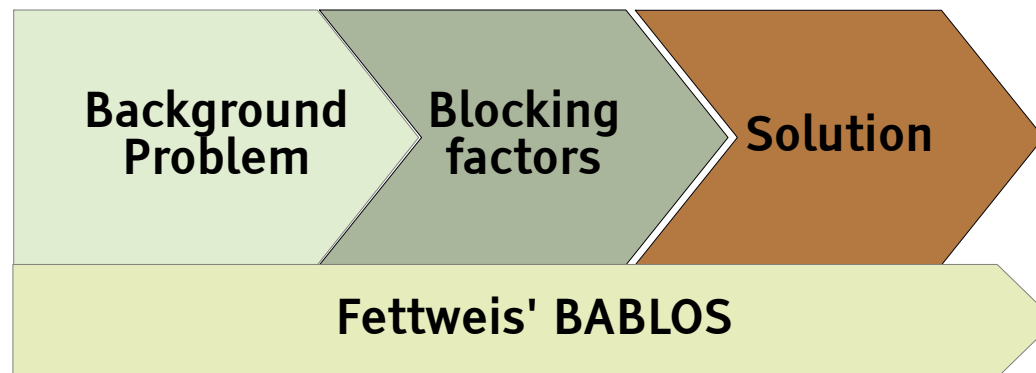
Studies of existing artifact-type have shown deficiencies on property. An enhanced design for an artifact-type is described, based on solution strategy. In comparison with existing solutions, it offers enhanced levels of property, according to analyses based on model-type. These improvements have been confirmed / demonstrated in tests of a working artifact-type based on the design.

ET: Enhanced tool

The effectiveness of model-type / solution strategy in supporting the design of artifact-type has been demonstrated. An enhanced tool / method is described for the design of artifact-type based on model-type / solution strategy. Examples are provided confirming the effectiveness of its support for model-type / solution strategy in design.

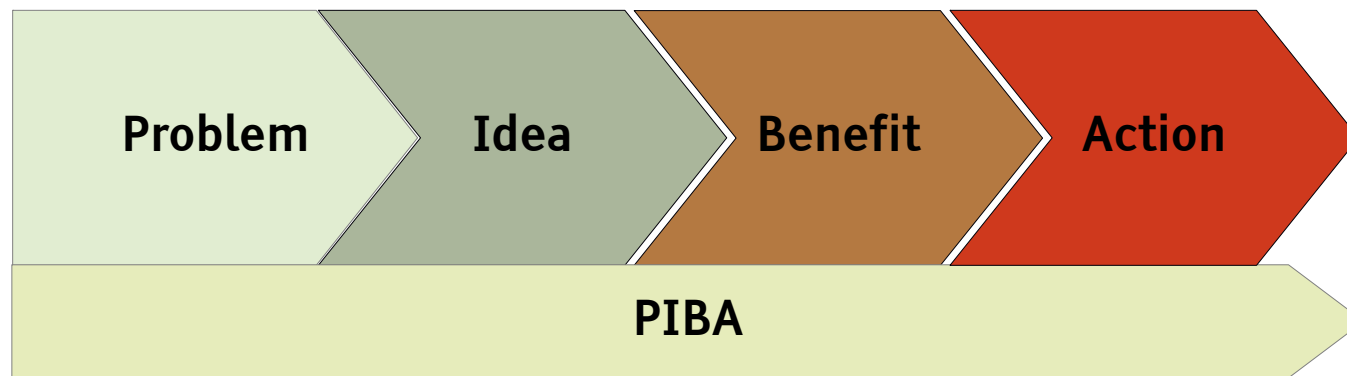
BABLOS 3-Step

- ▶ Gerhard Fettweis' Abstract Scheme: BABLOS-3-step
- ▶ (a) Background Problem of Context, Development, Society, Change
- ▶ (b) Blocking Factors, why no good solution exists
- ▶ (c) Solution



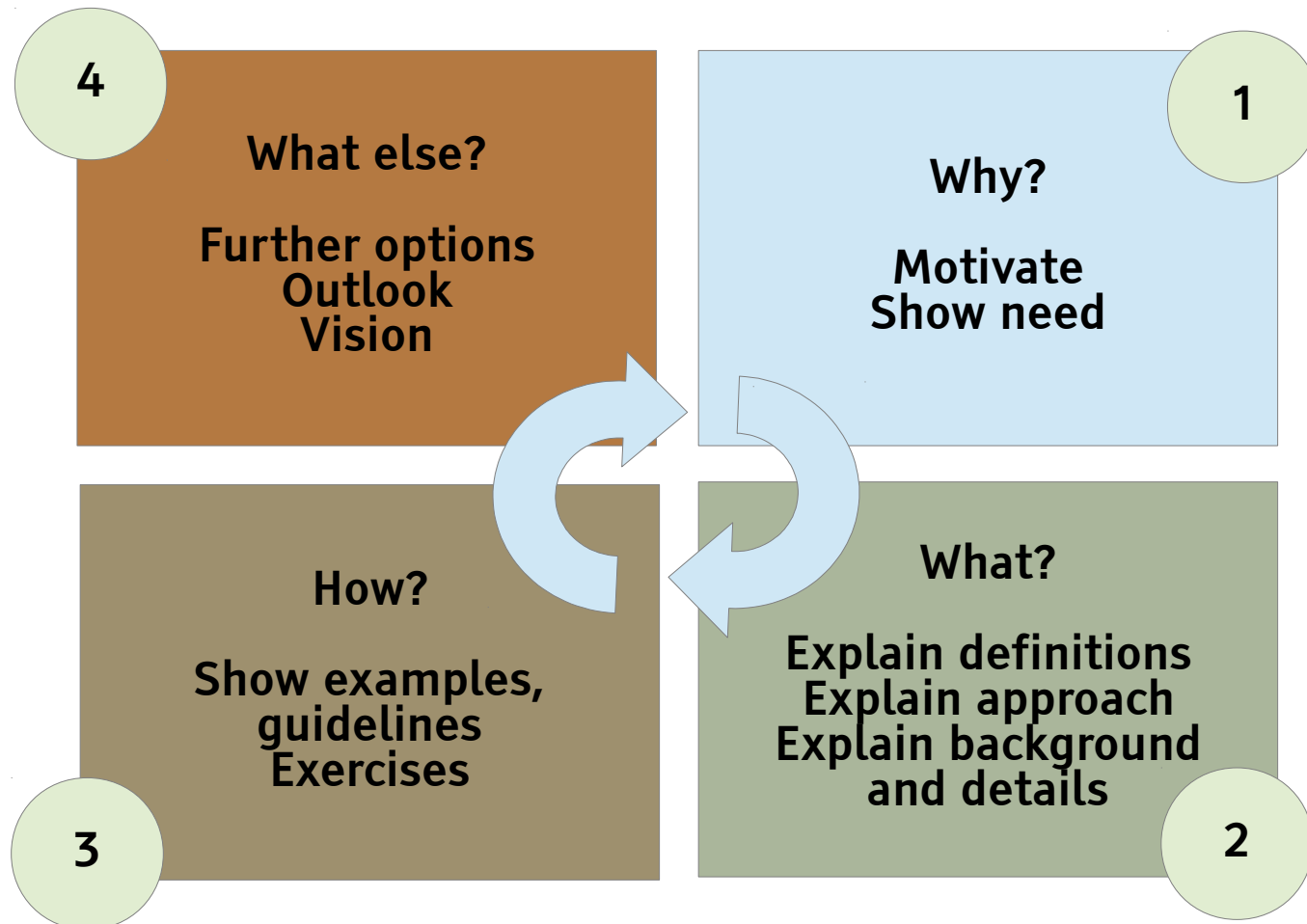
PIBA 4-Step Benefit Analysis

- ▶ Rombach and others invented PIBA as part of GQM+Strategies. It is a very simple scheme to derive actions from a beneficial idea
- ▶ (a) Problem of Context, Development, Society, Change
- ▶ (b) Idea
- ▶ (c) Benefit
- ▶ (d) Action



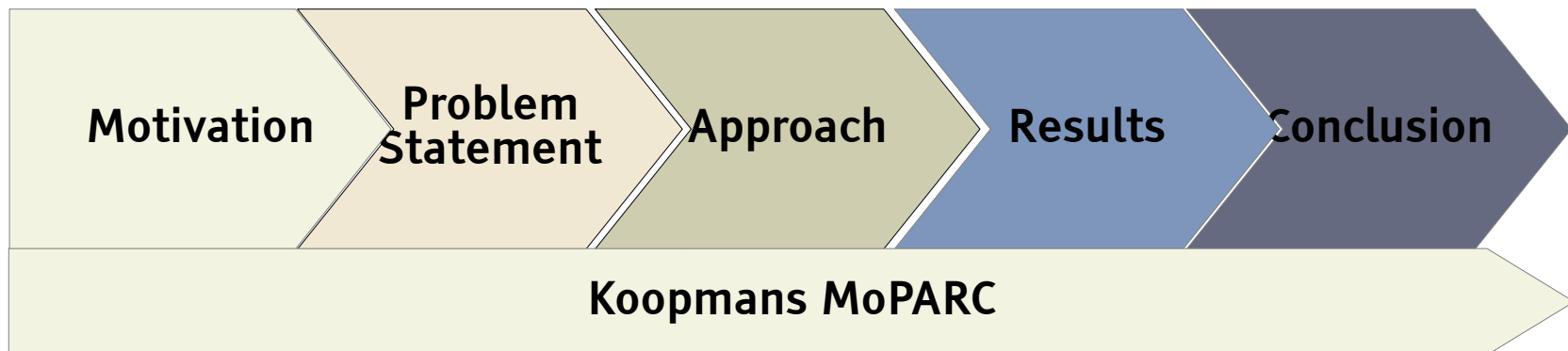
4-Quarter-Cake of Vigneshow/Schneider/Meyrose

- ▶ [Vigneshow-Schneider-Meyrose] 4-Step for Abstracts, Talks, Essays
- ▶ based on 7 honest serving men:
 - Cause → Result → Realization → Vision



MOPARC-Scheme of Philip Koopman

- ▶ www.ece.cmu.edu/~koopman/essays/abstract.html
- ▶ The MOPARC-Scheme of Philip Koopman “How to write an Abstract” is a 5-step scheme
 - Differs from PROBLOS leaving out the goals and blocking factors
 - Emphasizes results



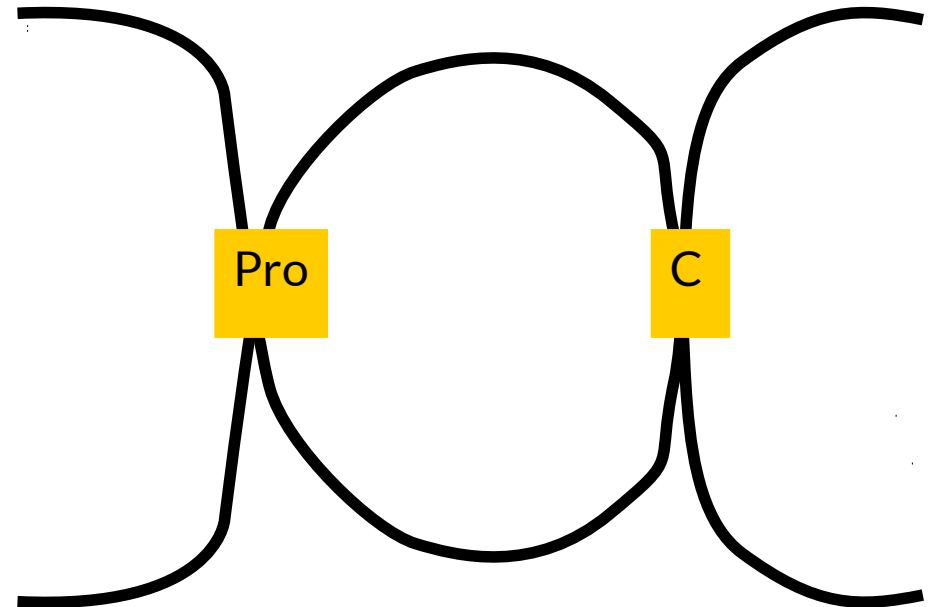
Kasper Österbye's Gul Caramel Christmas Cracker - Double-tailed fish MoProSoCO

41 Academic Skills in Computer Science (ASICS)

- ▶ A 5-step for an essay, a PhD report, an abstract, a conclusion chapter
- ▶ Similar to MOPARC of Koopmans, but with outlook
- ▶ Results are hidden in conclusion

MoProSoCO

1. Motivation
2. Problem
3. Solution (my own work)
4. Conclusion
5. Outlook



Overview of Template Abstracts

Problem-oriented development schemes for abstracts	Hint
PIBA	Simple action-oriented scheme; no introduction of approach
4-quarter cake	Easy to remember, not as complex as the 6 honest serving men
ZOPP	if success criteria play a role: very good for a Master's or PhD thesis
BPOPP	if blocking factors shall be highlighted
BATE-BPOPP	if background and technical problems shall be distinguished; very good for introductions of books, where different classes of readers are expected
MOPARC	Plain abstract scheme for research papers
Gul Caramel MOPROSOCO	Contains a reflection about the result
NABC	Need – Approach – Benefit for Cost - Competition

Exercise

- ▶ Write an abstract with MOPARC on the following theses:
 - The Higgs Boson found in 2012 is the particle giving mass to other particles.
 - The natural energy paradigm requires more long-distance power lines.
 - In 2012, Germany reached the threshold of 25% of energy production from natural sources.
 - After a long time, in 2012 Olympics was organized in London again.
- ▶ Write an abstract with BPOPP on the following theses
 - Japan wants to switch off nuclear power plants, but suffers from a shortage of energy then.
 - The size of the Sahara is increasing every year.
 - The arctic ice area turns from a permanently frozen zone to a instably frozen zone.

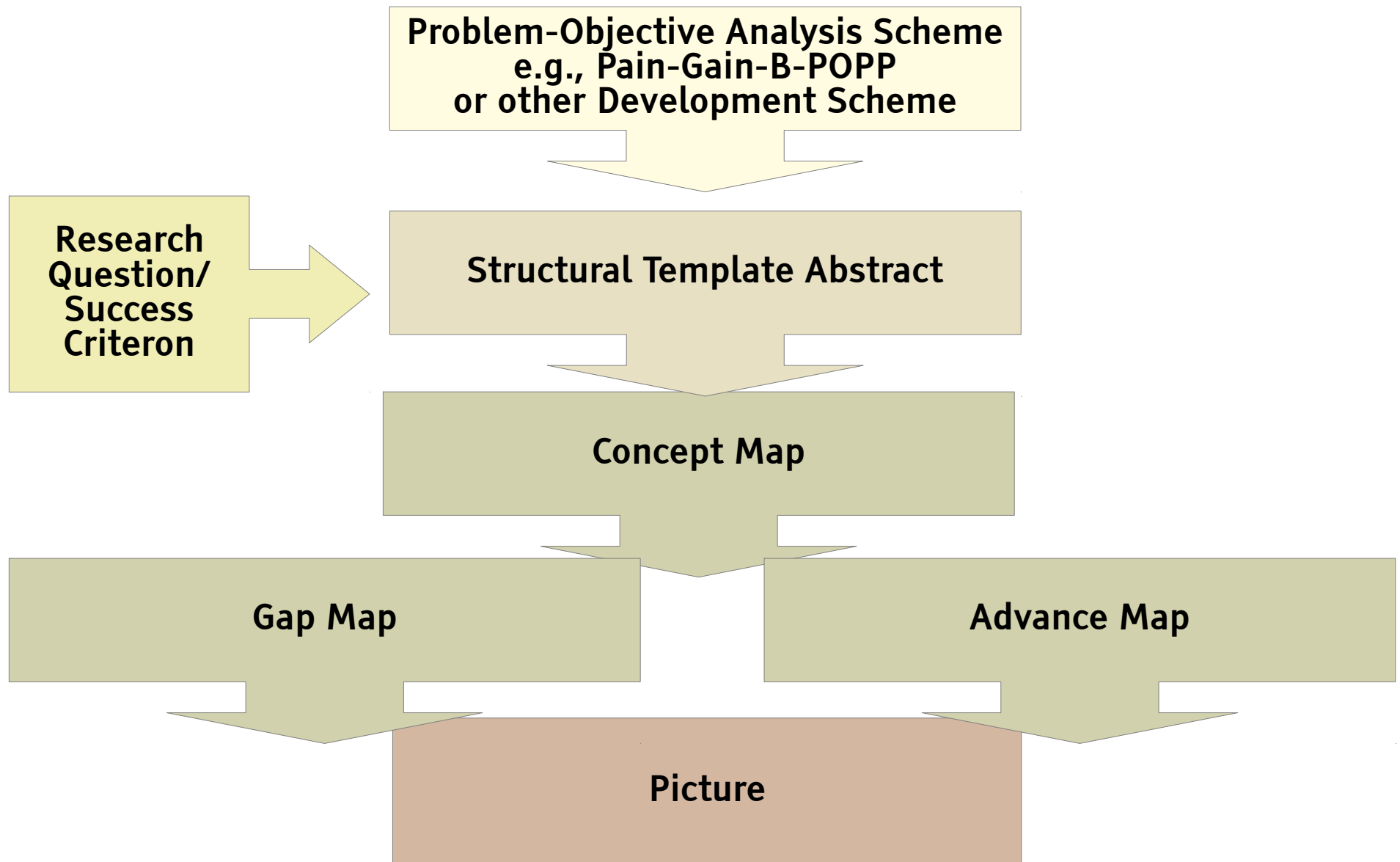
Exercise

- ▶ Write abstracts for the following thesis statement, with BABLOS, PIBA, ZOPP, nABC and GulCaramel. Compare!
 - **In 2022, Germany wants to shut down its last nuclear power plant, but there are many problems.**

31.4 Concept Maps Summarizing the Abstract and the Paper

Concept mapping for papers

- ▶ For every paper, a concept map should be produced



31.5 The “Discussion” Part

- ▶ The “discussion” is an answer to the form of the research hypotheses

A Discussion Treats Internal and External Aspects

- ▶ Apart from the validation part, a paper needs to have a **discussion part**

- ▶ The discussion part needs to emphasize several **internal aspects** discussed before:
 - **Advantages and Benefits**
 - **Disadvantages and Costs**
 - **Limits** (real limits, scope and assumptions)
 - **Open questions**
 - The discussion of the internal aspects is guided by the research hypotheses

- ▶ Also **external aspects** should be covered (subsection “Comparison to Related Work”)
 - **Unique selling points** other research results do not have
 - **Key performance indicators (metrics)** and how they could be improved

31.6 Different Kind of Research Results

What You Can Expect from a SE Researcher

- ▶ Remember the difference of engineers and technical scientists:
 - An engineer works out systems to solve problems
 - a technical scientist works out methods and techniques for engineers
- ▶ Papers (examples):
 - Problem papers
 - Literature analysis studies
 - SWOT analyses (strategic analyses)
 - Solution Pattern descriptions/papers
 - HOWTO-Papers (methods, process patterns)
 - Design pattern papers
- ▶ Artefacts (demonstrators often in 1st, 2nd and 3rd generation, most often not for industrial use):
 - Code Libraries and Frameworks helping other people doing work
 - Model frameworks
 - Tools for automation, for specific languages
 - Composition systems and reuse languages
 - Interpreters and compilers for languages
 - Books overviewing a subject area or method

The Daily Schmidt

- ▶ Read “Er war ein Gegner, nicht ein Feind – Laudatio über Karl Theodor Freiherr von und zu Guttenberg (1972)
- ▶ <http://www.zeit.de/1972/41/er-war-ein-gegner-nicht-ein-feind/komplettansicht>
- ▶ What is the thesis of the essay?
- ▶ How would you write a laudatio about a person of your choice?

What You Can Expect from a SE Researcher

- ▶ Remember the difference of engineers and technical scientists:
 - An engineer works out systems to solve problems
 - a technical scientist works out methods and techniques for engineers
- ▶ Papers (examples):
 - Problem papers
 - Literature analysis studies
 - SWOT analyses (strategic analyses)
 - Solution Pattern descriptions/papers
 - HOWTO-Papers (methods, process patterns)
 - Design pattern papers
- ▶ Artefacts (demonstrators often in 1st, 2nd and 3rd generation, most often not for industrial use):
 - Code Libraries and Frameworks helping other people doing work
 - Model frameworks
 - Tools for automation, for specific languages
 - Composition systems and reuse languages
 - Interpreters and compilers for languages
 - Books overviewing a subject area or method

The End

Mary Shaw: “A research paper is a purposeful, designed artifact, just like a software system. Apply software design techniques to paper design:

- ▶ *Start with the requirement:* read the call for papers
- ▶ *Select an architecture:* plan the sections, what they say
- ▶ *Plan a schedule:* allow time for review, revision
- ▶ *Check consistency:* type-check text like code”