Theory Relationships in D and E

Dewayne E Perry
Dept. of Electrical and Computer Engineering
The University of Texas at Austin
Austin TX, USA
perry@mail.utexas.edu

***Abstract*—My general theory of software engineering is that there are two logical parts: design and evaluation. In my theories D and E (Design and Evaluation) there are references to a theory T in each of them. The interesting question is "what is the relationship between these two theories D.T and E.T. I first delineate a rich variety of theories related to D.T, D.M, and E.T and consider these to be sub-theories critical to understanding the relationships between D.T and E.T. I then delineate these relationships in the context of different types of empirical evaluations. The resulting delineation helps to explain why empirical evaluations should be a dominant part of software engineering projects.**

***Index Terms*—Theories in Software Systems, Theories in Empirical Evaluations, Relationships among Theories**

I. Introduction

In my earlier work [1][2][4], I introduced my basic theory of software engineering as consisting of two logical parts: design and empirical evaluation. I have proposed two simple theories, *D* and *E*, as the basis for laying out a unified theoretical foundation for software engineering and software engineering research. Theory D is the theoretical basis for the design part, and theory E is the theoretical basis for the empirical evaluation part. Models for D and E are defined, using my theory modeling language, and subsequently, the theories D and E are composed using my model calculus in various ways, and the composed models are used to explore various ways of thinking about the underlying foundations for software engineering and software engineering research.

The terms "theory" and "model" are used and misused in a variety of ways, often informally and interchangeably. They are used here in a very specific way: a theory (a more or less abstract entity) is reified, represented, satisfied, etc., by a model (a concrete entity). This view of theories and models is derived in part from Turski and Maibaum [5] where they state "*A specification is rather like a natural science theory of the application domain, but seen as a theory of the corresponding program it enjoys an unmatched status: it is truly a postulative theory, the program is nothing more than an exact embodiment of the specification*".

In my approach, theory is considered to be broader than a specification and, more than likely, less formal. Logically, a model is an interpretation of a theory and has certain logical properties. In my approach, I broaden the notion of a model to be a representation (indeed, a reification) of a theory. The model is of paramount importance in design disciplines as it is the visible manifestation of a theory. Of fundamental importance is the fact that a theory can have an arbitrary number of models.

Theory D is meant to capture the typical cycle of creating a theory that is then reified into a model where the model is then injected into the world – i.e., an initial, non-iterative, non-evolutionary development of a model from a theory. The injection of the model into the world results in a changed world[1]. D is summarized as follows:

- We observe and abstract some specific part of the world and create a theory.
- From that theory we create a usable model to reify or represent that theory.
- When satisfied that the model adequately represents the theory, we inject the model into the world.

The model of D, representing the informal theory above, consists of three elements (objects) and three transformations (mappings, or, if you will, processes). The elements are as follows:

- **W** – the world, but more specifically, the part of the world relevant to the theory;
- **T –** the theory initiated by observations, selections, and abstractions; and
- **M –** a model that reifies, represents or satisfies the theory T.

The transformations involving these elements of the model are as follows:

- **W → T –** generate a theory: observe, select, and abstract from the world W to create a theory T;
- **T → M –** from the theory T create a model M; and
- **M*W → W** – inject model M into the world W changing the world W.

Theory E is meant to capture the process of empirical evaluation. It is sufficient at this point to indicate that empirical evaluations can range from very informal evaluations (as indicated by this formulation of E) to formal and controlled experiments.

---

[1] Note, however, the world can also change for a variety of other reasons. The discussion of those reasons is beyond the scope of this paper.

Not surprisingly, the theory E is essentially a simplification of basic empirical science.

- Given a world W, observe, select, and abstract to create a theory
- Given a theory T, generate an hypothesis H to test some part of the theory
- From the hypothesis H, generate an evaluation regimen R.
- Apply the regimen R to the world W and revise theory T if necessary.

Note that this is a very basic theory, but it still is sufficiently rich to cover the entire range of studies from exploratory through to rigorously explanatory studies. In practice, theory T may be vague and ill-formed (as it would be for exploratory work) or well-formed and mature (as it should be when doing predictive or explanatory work). Similarly the hypothesis may be generic and open-ended or focused and specific. Evaluations E may be opportunistic (for exploratory work) or specific and well-designed. Further, the theory of E supports both theory generation (in the case of exploratory work) and focused theory evaluation.

The basic elements in the model are: world **W**, theory **T**, hypothesis **H**, and evaluation regimen **R**.

The following transformations represent the processes of conducting an empirical study.

- **W → T –** generate a theory T by observing and abstracting from the world W
- **T → H** – derive an hypothesis H from theory T
- **H → R –** create an appropriate evaluation regimen R based on H
- **R * W → T** – apply the regimen to the world and reconcile theory and reality – i.e., on the basis of the evaluation and the current theory T, revise T if necessary.

While I expand both of these theories and models in [4], they are sufficient for the purpose needed in this discussion.

Note that these theories (represented in the two models by T) are a part both D and E. The interesting question I want to address in this paper is: are D.T[2] and E.T related and, if so, how?

II. Different Types of Theories

I consider the design part of software engineering to be a model of theory D and the evaluation part of software engineering to be a model of theory E. It is in this context that the possible relationships between D.T and E.T are important.

In the design part of software engineering, D.T is the set of requirements used to build and evolve D.M, where D.M is the software system. In the evaluation part of software engineering, E.T is the theory used to evaluate various aspects of both D and E (when evaluating how good the evaluation is) in the theory and model compositions E:D and E:E.

At a theoretical level, one can make the case that there are a large number of theories D.T that are all more or less unique representing different types of systems in differing domains. In this case then, one can think of theories in terms of domains as a means of distinguishing different aspects of theories.

But it is also useful in software engineering requirements (i.e., D.T) to distinguish between functional and non-functional requirements. At a more abstract theoretical level, this is a distinction between behavior and constraints on that behavior (for example, performance, reliability, etc.).

One can make similar separating distinctions in D.T depending on the type of evaluation being done. For example, in the evaluation of D.M one can distinguish between evaluating architectural, or structural, aspects of D.M from that of the functional aspects. In evaluating evaluations, there are different kinds of validity to evaluate (for example, construct, internal, statistical, and external validity). Similarly, there will be different evaluation theories for evaluating the various transformations of both D and E.

In the rest of the paper, I will explore these various distinctions that can be made and how they are related to D.T and E.T. Specifically, I will delineate the following types of theories:

- about theories in D;
- about models in D;
- about transformations in D;
- about evaluating theories ;
- about evaluating models; and
- about evaluating evaluations.

Additionally, please note that the delineation of relevant theories and evaluations are meant to be thorough and illustrative but not necessarily completely comprehensive.

III. Types of Theories in D

The previous section mentioned three different aspects of D.T that are important: domains, behaviors, and constraints on behaviors. I believe it is useful to think of D.T in terms of these three aspects.

- Let T_B represent that part of D.T that delineates the desired behavior that the model D.M is to reify and satisfy.
- Let T_C represent that part of D.T that delineates the desired constraints on that behavior that the model D.M is to reify and satisfy.
- Let T_D represent the various domain theories that underlie the behavior and constraints on behaviors in D.T.

Typically T_C includes such constraints as performance constraints on certain behaviors, reliability constraints, security constraints, etc.

While we often think of a theory and its models as being of a single domain, the reality is that, while there may be a dominant, high-level domain (such as client-server, etc.), the

[2] I use the dot notation to reference specific objects in a particular model: D.T is the object T in model D and E.T is the object T in model E – that is, the two theories T that are the main subject of this paper.

reality is that there are a variety of sub-domains that are reified in the theory and models as well.

It is my theory that D.T is the union of T_B, T_C, and T_D and that each of these theory types will play a critical role relative to E.T in various types of evaluations of D.M.

IV. Types of Theories about Theories in D

There are a number of critical theories about theories in D that need to be considered and used in evaluating D.T. These theories are needed to determine if D.T is sufficient to effectively create a model D.M for D.T, whether D.T is consistent, whether D.T is complete, and whether D.T is feasible. These types of theories are all focused on determining whether D.T can be effectively used to create a model D.M that reifies and satisfies D.T.

- Let T_{TS} represent a theory about the sufficiency of the theory D.T.
- Let T_{TC1} represent a theory about the consistency of the theory D.T.
- Let T_{TC2} represent a theory about the completeness of the theory D.T.
- Let T_{TF} represent a theory about the feasibility of the theory D.T.

It is my theory that T_T is the union of T_{TS}, T_{TC1}, T_{TC2}, and T_{TF}, and that each of these theories will play a critical role in the evaluating D.T.

V. Types of Theories about Models in D

There are a number of aspects about D.M that are important besides the satisfaction of the theoretical *behavior* and attendant *constraints* of D.T. The *structure* of D.M provides the framework for reifying and satisfying D.T. From a user's point of view, the *interface* of D.M is a critical feature of usability. From the customers' standpoint, its *usefulness* (i.e., how useful it is in the context of the customers' needs) ultimately determines how satisfied they will be with the model D.M. Two further types of theories are needed to round out the theories about models: theories about model qualities, and theories of model metrics.

A. Behavior and Constraints

While a model evaluation considers the model M as a single entity, in practice D.M is comprised of a set of modules each with their own sets of behaviors and constraints.

- Let $T_{Bm}{}^i$ represents a behavior included in T_B that m^i is to reify and satisfy.
- Let $T_{Cm}{}^i$ represents a behavioral constraint included in T_C that m^i is to reify and satisfy.
- Let $T_{Dm}{}^i$ represents a domain included in T_D that underlies the behavior and constraints on behaviors in m^i.

T_B is included in the union of all the $T_{Bm}{}^i$s (the components of M together may in fact provide more behaviors and constraints that is needed to reify and satisfy its theory). The same is the case for T_{CM} and T_{DM}.

It is my theory that module-related behavior, behavioral constraint, and domain theories about the components of D.T will play a critical role in various types of evaluations of D.M.

B. Theories of Model Structure

Further, in practice D.M is comprised of a set of modules each with their individual structure and together are used to create the structural architecture of the system.

- Let T_{SM} represent the set of individual theories about the structures of the individual components, or modules, of D.M.
- Let $T_{Sm}{}^i$ represent a theory about the structure of an individual module m^i of D.M.
- Let T_{SA} represent the theory about the architectural structure of D.M.
- Let $T_{Sa}{}^i$ represent a theory about the (sub)architecture of a composed module a^i of D.M.
- Let T_S represent the union of T_{SM} and T_{SA}.

It is my theory that the structural theories about D.M will also play a critical role in various types of evaluations of D.M.

C. Theories of Interface Usability

Another element of D.M is the interface presented to the user (whether human or automaton) and how usable it is. This interface is comprised of two parts: the usability of *features* the user can invoke and the usability of *results* that these features provide.

- Let T_{IF} represent the set of individual theories about the usability of features of the interface of D.M.
- Let $T_{If}{}^i$ represent a theory about the usability of feature f^i of the interface of D.M.
- Let T_{IR} represent the set of individual theories about usability of the results of the interface of D.M.
- Let $T_{Ir}{}^i$ represent a theory about the usability of the results r^i of the interface of D.M.
- Let T_I represent the union of T_{IF} and T_{IR}.

Again, it is my theory that the usability theories about the interface of D.M will play a critical role in the various types of evaluations of D.M.

D. Theories of Usefulness

Ultimately, customers decide how useful a model /system is with respect to their needs. To avoid building useless systems, we need a surrogate with which to valuate usefulness. Thus, let T_U be a theory about the usefulness of D.M. It too plays a critical role in the various types of evaluations of D.M. If needed, we can mirror the theories of usability and consider theories of usefulness for individual features and results.

They too will play a crucial role in the various types of evaluations of D.M.

E. Theories about Model Qualities

Often categorized with theory behavior constraints, these qualities about the models are not constraints on behavior, but constraints on various aspects of the models independent of their behavior. These constraints on the models include such things as understandability, style conformance, maintainability, changeability, etc. They are distinct from constraints on behaviors in the theories (requirements).

- Let T_Q be a general theory about the quality constraints on a Model D.M.
- Let T_{QU} be a specific theory about the understandability of a model.
- Let T_{QS} be a specific theory about the (coding) style conformance of a model
- Let T_{QM} be a specific theory about the maintainability of a model.
- Let T_{QC} be a specific theory about the changeability of a model.
- Etc., as desired.

T_Q is the union of T_{QU}, T_{QS}, T_{QM}, T_{QC}, etc.

It is my theory that the quality theories about D.M will also play a critical role in various types of evaluations of D.M.

F. Theories about Model Metrics

There are a variety of metrics that are possible to use in characterizing the model D.M. Among the most common metrics are the number of faults, complexity, cohesion, coupling, and amount of cloning present. Let T_M be the general theory of model metrics which is comprise of at least the following theories of individual metrics.

- Let T_{Mf}^i represent the theory about faults in component i.
- Let T_{MF} represent the set of faults found in all the components in model D.M.
- Let T_{Mc1}^i represent a theory about the complexity in component i.
- Let T_{MC1} represent a theory about the complexity of the model D.M.
- Let T_{Mc2}^i represent the theory about cohesion in component i.
- Let T_{MC2} represent the theory of cohesion for model D.M.
- Let T_{Mc3}^i represent the theory about coupling in component i.
- Let T_{MC3} represent the theory of coupling in model D.M.
- Let T_{MC4} represent the theory of cloned code in model D.M.
- Let T_{Mc5}^i represent a theory about code coverage in component i.
- Let T_{MC5} represent a theory of code coverage in model D.M.

It is my theory that the metric theories about D.M will also play a critical role in various types of evaluations of D.M.

VI. Types of Theories about Processes

The transformations listed in the simple models of theories D and E represent the design (theory reification) processes and evaluation processes respectively. Akin to the various types of theories we have about models, we also have various types of theories about these processes, such as best practices, process improvement, process metrics, etc.

- Let T_P be a general theory about processes for D and E.
- Let T_{Pb}^i be a specific theory about the best practice for a process in D or E.
- Let T_{PB} be the union of theories about best practices for the individual processes in D and E.
- Let T_{Pi}^i be a specific theory about process improvement for a process in D or E.
- Let T_{PI} be the union of theories about process improvement for the individual processes in D and E.
- Let T_{Pm}^i be a specific theory about process metrics for processes in D or E.
- Let T_{PM} be the union of theories about the process metrics for the processes in D and E.
- Etc., as desired.

T_P is the union of T_{PB}, T_{PI}, T_{PM}, etc.

Process metrics include a wide variety of issues such as effectiveness (often measured in terms of faults introduced, and faults removed), and productivity (in terms both time and cost).

It is my theory that the transformation theories for the processes in D and E will also play a critical role in various types of evaluations of D and E.

VII. Types of Theories about Evaluating Theories

Theories about evaluating theories include at least the following: peer reviews about theories (both of theory segments and theories in its entirety), and analysis of theories.

- Let T_{ER1} represent a theory about evaluating theories by means of peer reviews.
- Let T_{EA1} represent a theory about evaluating by means of analysis.

It is my theory that these theories about evaluating theories will also play a critical role in various types of evaluations of D.

VIII. Types of Theories about Evaluating Models

We have a rich set of related theories about the evaluation of models that range from evaluations to ensure that a model sufficiently reifies and satisfies the related theory, to evaluations to ensure that the model has a given set of desired structural properties.

- Let T_{ER2} represent a theory about evaluating models by means of peer reviews.
- Let T_{EA2} represent a theory about evaluating models by means of analysis.
- Let T_{EU} represent a theory about evaluating models by means of unit testing.
- Let T_{EI} represent a theory about evaluating models by means of integration testing.
- Let T_{ER3} represent a theory about evaluating models by means of regression testing.
- Let T_{ES} represent a theory about evaluating models by means of system testing.
- Let T_{EL} represent a theory about evaluating models by means of load testing.
- Let T_{EA3} represent a theory about evaluating models by means of acceptance testing.

It is my theory that these theories about evaluating models will also play a critical role in various types of evaluations of D

IX. Types of Evaluation^2 Theories

Analogous to evaluating theories in D is evaluating theories in E. These criteria are the same in both cases.

- Let T_{ES} represent a theory about the sufficiency of the theory E.T.
- Let T_{EC1} represent a theory about the consistency of the theory E.T.
- Let T_{EC2} represent a theory about the completeness of the theory E.T.
- Let T_{EF} represent a theory about the feasibility of the theory E.T.

It is my theory that T_T is the union of T_{TS}, T_{TC1}, T_{TC2}, and T_{TF}, and that each of these theories will play a critical role in the evaluating E.T.

Similarly, we need theories T_H about evaluating hypotheses E.H and T_R about evaluating the regimen that takes a subject and manipulates. In the discussion below on the relationships between the theories of D and E, we will see that sometimes theories in or about D will be used as both E.T and as the subject in regimen E.R.

One of the critical aspects of evaluations is the determination of how good and reliable the evaluations are (referred to here as evaluation^2 – i.e., evaluation of an evaluation). Analogous to faults in software systems, validity issues are the determinants of the quality of empirical evaluations and the strength or dependability of their results.

- Let T_{VC} be a theory about the construct validity of an evaluation.
- Let T_{VI} be a theory about the internal validity of an evaluation.
- Let T_{VS} be a theory about the statistical validity of an evaluation.
- Let T_{VE} be a theory about the external validity of an evaluation.

Whether all of these theories are needed in evaluating and evaluation depend on what type of evaluation it is. If it is a predictive evaluation, then there is no need to evaluate the study for external validity since it is the confirmation or denial of the prediction that is of critical importance. Case studies do not need to be evaluated for statistical validity as the reasoning about case studies is not a statistical enterprise. Construct and internal validity are always critical to any study: if we don't have good abstract and concrete constructs (instruments and measures), and we don't rule out confounding variables and alternative explanations, our studies are weakened and possibly completely useless.

X. E.T and Various Evaluations

The overall role of empirical evaluations in software engineering is to answer critical questions about the various aspects of the creation and evolution of the desired software system. As such, empirical evaluations are the drivers of the micro-evolutionary iterations of the software system from beginning to end (i.e., to release).

Because of this driving nature of evaluations, it is critical to understand the relationships among the various types of theories and the various elements in those theories and models.

A. Evaluation of Theories

In the evaluation of theories, the theories are the subject of the evaluations as described in the hypotheses and the regimens. In the context of D.T, T_B, T_C, and T_D are all covered as subjects in the theory evaluations performed in software engineering.

As noted so far in this paper, we have a very large variety of theories that are delineated and used the evaluations of D.T. These theories delineated here are in fact, the subjects of theory evaluation in software engineering research. But I consider the discussion and evaluations of these theories to be the topic of a separate paper.

1) Peer Reviews

Peer reviews are typically an informal and exploratory empirical evaluation focusing, at least in the beginning, on small segments of D.T, and later on the complete theory, D.T. In general, E.T is T_T (a theory about theories) and T_{ER1} (a theory about evaluating theories by means of peer reviews).

In the early stages of evaluating D.T the focus may well be on the sufficiency and feasibility of D.T since the issues of consistency and completeness make more sense later in the process of creating and evolving D.T. In this case, E.T would be either T_{TS} or T_{TF}, or both.

In the later stages of creating and evolving D.T, the issues of completeness and consistency are of paramount importance. In that case, E.T could be T_T. We note, however, that in many peer review processes, the various reviewers are given specific roles to play in the review. In this case, the various types of theories about theories may be assigned to different reviewers and, at least for the individual evaluations, E.T may be one of T_{TS}, T_{TC1}, T_{TC2}, or T_{TF}.

2) Analysis

Since sufficiency and feasibility require human judgment, they are unlikely to be amenable to analysis even if the theory is formally described. If the theory is formally described in a form that can be automatically analyzed, then analysis is possible and E.T is T_{TC1}, T_{TC2}, and T_{EA1}.

B. Evaluations of Models

The two standard forms of model evaluation used for all software engineering systems are code reviews and testing. Peer reviews are typically done in code segments that may or may not coincide with single or multiple individual components of a model. Obviously, D.M is the subject of the evaluation and of the manipulations in the regimen E.R.

Testing begins with the testing of individual components of the model, moves through testing of increasingly larger collections of components in integration testing, and then a variety of complete model testing efforts. In the latter case, D.M is clearly the subject of the evaluation, of E.H, and of the manipulations in E.R and in the earlier stages the subjects of the evaluations are components of D.M.

Just as D.T has a model D.M, so too do the theories delineated in the paper often have models used to reify and satisfy these theories. Again, the evaluations of these models are, as with their corresponding theories, the province of software engineering research and are the topic of a separate paper.

Using the theories that comprise E.T predictively, the failures of those predictions generally lead to changes in the corresponding models. Hence the need for iteratively evaluating both the model itself, but also the various components of those models.

1) Peer Reviews

Peer reviews of models, as with peer reviews of theories, are typically an informal and exploratory empirical evaluation focusing on small segments of D.M. The comprehensive evaluations of models are done using various different aspects testing evaluations. In general, the basic part of E.T is T_M (a theory about models) and T_{ER2} (a theory about evaluating models by means of peer reviews). However, E.T is far richer than just T_M and T_{ER2}. E.T also includes the following theories:

- Elements of D.T – that is, $T_{Bm}{}^i$, $T_{Cm}{}^i$, $T_{Dm}{}^i$ – since only portions of the model (code) are reviewed in an individual review;
- Structural issues for the portions of the model: $T_{Sm}{}^i$ and possibly $T_{Sa}{}^i$ if the review covers multiple model components;
- Usability issues for the portions of the model: $T_{IF}{}^i$ and $T_{IR}{}^i$;
- Usefulness issues for portions of the model: $T_{Um}{}^i$;
- Model quality issues as exemplified in the various types of theories in T_Q; and
- Model metrics issues as exemplified in the various types of theories in T_M.

Thus, evaluating models by means of peer reviews involves a very rich set of theories that make up E.T. They cover a comprehensive set of issues concentrated in one form of model evaluation.

2) Analysis

Given that models are ultimately formally described (in code), analysis of models can be more precise and focused that peer reviews. In general, the basic part of E.T is T_M (a theory about models) and T_{EA2} (a theory about evaluating models by means of analysis). However, as we noted for peer reviews, E.T is far richer than just T_M and T_{EA2}. E.T also includes the following theories for partial analyses of models:

- Elements of D.T – that is, $T_{Bm}{}^i$, $T_{Cm}{}^i$, $T_{Dm}{}^i$ – when only portions of the model (code) are analyzed;
- Structural issues for the portions of the model: $T_{Sm}{}^i$ and possibly $T_{Sa}{}^i$ if the analysis covers multiple model components; and
- Model metrics issues as exemplified in the various types of theories in T_M.

Analysis is unlikely to cover theories of usefulness and usability as they tend to be emphasize qualitative aspects rather than quantitative. However, there is one aspect of usability that can be analyzed, namely, that the uses of features and results are proper and correct ones. Thus, we add theories $T_{IF'}$ and $T_{IR'}$ as elements of E.T.

In addition to the analysis of components or segments of D.M, it may be possible to analyze the system as a whole (at least at some level of abstraction). In this case, D.T is included in the mix of E.T as well as T_S.

As with evaluating D.M by peer reviews, evaluating D.M by means of analysis can involve a very rich set of theories as part of E.T.

3) Unit Testing

Unit testing is an empirical evaluation that focuses on evaluating individual components of D.M. The set of theories comprising E.T is much more restricted because of this focus. Unit testing is concerned both about whether the appropriate behavior, constraints, and domain theories relevant to that component are reified and satisfied. In addition, unit testing is concerned that the relevant structural theories for that component are satisfied as well. Thus we have the following theories as elements in E.T:

- T_M (a theory about models);
- T_{EU} (a theory about unit testing);
- $T_{Bm}{}^i$, $T_{Cm}{}^i$, $T_{Dm}{}^i$ relevant to that component of the model D.T;
- $T_{Sm}{}^i$ relevant to that component of the model D.M; and
- $T_{Mc}{}^i$ – the theory of code coverage for that particular component.

Unit testing is a predictive evaluation in which E.H and E.R predict certain results for both theory satisfaction of elements of E.T and structural coverage of the component.

4) Integration Testing

Integration testing is an empirical evaluation that focuses primarily on the interactions of components of D.M but as with unit testing it also includes the relevant behavior, constraint, domain, and interface theories for the components being integrated, as well as the appropriate structural theories.

- T_M (a theory about models);
- T_{EI} (a theory about integration testing);
- $T_{Bm}{}^i$, $T_{Cm}{}^i$, $T_{Dm}{}^i$ relevant to each of the components of the model D.M;
- $T_{Sm}{}^i$ relevant to each of the components of the model D.M;
- $T_{Sa}{}^i$ relevant to relevant architecture structure of the model D.M;
- $T_{IF'}$ and $T_{IR'}$ for the relevant features and results of this integration; and
- $T_{Mc}{}^i$ – the theory of code coverage for that particular component.

Integration testing is also a predictive evaluation in which E.H and E.R predict certain results for both theory satisfaction of elements of E.T, certain interactions among the components of D.M, and structural coverage of the components.

5) System Testing

It is system testing that the main focus is on evaluating how well D.M reifies and satisfies D.T. So while there are other theories involved in E.T for system testing, D.T is the main focus of E.T. Basically E.T is D.T. But there are other theories about models that come into play as well:

- T_M (a theory about models);
- T_{EI} (a theory about system testing);
- T_U (a theory about usefulness); and
- T_I (a theory about interface usability).

One might also want to test the structural theory T_S and the coverage theory T_{MC5} about D.M.

Depending on the emphasis of E.H and E.R, the system test evaluations may be either predictive or exploratory.

6) Load Testing

Load testing is basically the same as system testing except for a more thorough evaluation of the theory about constraints, T_C, in D.T.

7) Acceptance Testing

Acceptance testing is the combination of system and load testing, except that is the actual users who are performing the evaluations. Not surprisingly, in addition to insuring that D.T is adequately reified in D.M, there is additional emphasis on the theory of interface usability T_I and the theory of usefulness T_U.

8) Regression Testing

Regression testing is an evaluation to insure that, despite changes made, the rest of the system is unaffected and D.M continues to satisfy D.T and the other model-related theories. The subjects of regression test range from components, to arbitrary collections of components, to sub-architectures, and to the complete model. Thus, in addition to TER3 as an element of E.T, all the theories encountered in the various testing evaluations may be included in E.T depending on what levels of testing and what components of D.M are included in the regression evaluations.

C. Evaluations of Processes

Evaluations of processes inherently induce process improvements. The scientific cycle of testing theories and reconciling theory with reality induces changes in theories, and changes in theories should be reflected in their reifying models.

These theories about the processes in D and E then become the subjects of the evaluations and explicitly are expressed in E.H and E.R.

There are two general approaches that have evolved in evaluating processes: using various types of metrics to evaluate the effects of processes and using these metrics to drive the improvement process; and using exemplars, here called best practices, and the ideals, and adopting various parts of these best practice theories as needed.

1) Process Improvement and Metrics

There are two classes of metrics that can be used in valuating processes and improving them: model metrics, as described above, that provide an indication of how good the processes are at producing good models; and process metrics that have been referenced but not described. To expand on the latter, typical process metrics include time, cost, and resulting quality (often measured in terms of faults, but may also be measured by the various product metrics as well). Time can be measured in terms of race time and/or lapse time.

Thus for a particular process E.T would include T_{Pi}^{i}, a collection of process metrics as needed, $T_{Pm}^{i} – T_{Pm}^{j}$, and any of the model metrics needed from T_M.

2) Best Practices

One form of process evaluation is comparing them to exemplars – in this case we refer to these exemplars as *best practices*. Theories about best practices come about through practitioner consensus, typically after repeated cycles of process evaluations and improvements as described above.

One then needs a theory of comparing theories as E.T with both the current process theory and the best practice theory as subjects.

Another form of process evaluation is to use the best practice theory, T_{Pi}^{i}, as E.T to predict characteristics to be found in the subject process theory. As with models, when the corresponding theory predictions fail, the subject theory needs to be changed.

D. Evaluations of Evaluations

As delineated above, there are a rich set of theories about various kinds of evaluations of both theories and models, but also about evaluating evaluations. The basic things to evaluate are the same in all cases: are the theories sufficient, consistent, complete, and feasible; and what are the strengths and weaknesses in terms of construct, internal, statistical, and external validity?

The theories evaluated then are the subjects of E.H and E.R and E.T is a combination of T_{ES}, T_{EC1}, T_{EC2}, and T_{EF} together with T_{VC}, T_{VI}, T_{VS}, and T_{VE}.

Of course these theories need to be evaluated as well.

Where tools are used as a means of evaluations, then in addition to evaluating their underlying theories, one must also evaluate how well those tools (i.e., models) reify and satisfy their corresponding theories. An evaluation tool is effective only to the extent that it correctly and sufficiently reifies its theory.

XI. Summary and Conclusions

As in the software engineering of software systems where the devil is in the details, so too the devil is in the plethora and types of theories to describe 1) the behaviors, constraints, and domains of D.T that are to be reified in the models D.M, 2) the structure and character of the models D.M, 3) the structure and character of transformations (processes) in both D and E, and finally 4) the theories about character of evaluating evaluations.

So not only are our software systems inherently complex, a general theory of the software engineering of these systems is also inherently complex. What I have laid out here are just theoretical types and interrelationships for the technical part of the software engineering enterprise. I believe that the issues encountered here are justification for the theory structure we propose in [3].

I have not even touched on the theoretical issues of project management and project structure, nor on the theoretical issues relative to the software engineers who design and evaluate these complex systems.

Nor have I touched the theoretical issues concerning software engineering research – though the theoretical issues discussed here will be relevant to that discussion, but at a higher level of abstraction. This is one my next topics on my list of future work.

The other critical results of this paper are 1) the centrality of theories and the relationships in the two elements of software engineering, design and evaluation, and 2) the fundamental importance of empirical evaluation of both of these elements.

Acknowledgment

First, thanks to the SEMAT workshops for supporting this critical and timely topic of research. Second, thanks to Don Batory for his willingness to listen, patience, comments, and suggestions in the development of my research approach.

References

[1] Dewayne E Perry. "A Unifying Theoretical Foundation for Software Engineering", 20th International Conference on Software Engineering and Data Engineering (SEDE), June 2011.

[2] Dewayne E Perry, "A Theoretical Foundation for Software Engineering: A Model Calculus", 2nd Semat Workshop on a General Theory of Software Engineering (GTSE 2013), 35th International Conference on Software Engineering (ICSE 2013), San Francisco, May 2013.

[3] Dewayne E. Perry and Don Batory, "A Theory about the Structure of GTSEs, January 2015. Submitted for publication.

[4] Dewayne E Perry, "A Framework for Exploring Unifying Theories of Empirical Software Engineering", October 2014. Accepted for publication in the Science of Computer Programming, 2015.

[5] Turski, Wladyslaw M. and Maibaum, Thomas S. E. *The Specification of Computer Programs*. Reading, Mass: Addison-Wesley,1987