Project Control System to Track and Optimize Chip Design Projects

Stefan Häusler, Jana Blaschke, Christian Sebeke, Wolfgang Rosenstiel and Axel Hahn

***Abstract* — Nowadays, the managing of product development projects is increasingly challenging. Especially the IC design of ASICs with both analog and digital components (mixed-signal design) is becoming more and more complex, while the time-to-market window narrows at the same time. Still, high quality standards must be fulfilled. Projects and their status are becoming less transparent due to this complexity. This makes the planning and execution of projects rather difficult. Therefore, there is a need for efficient project control. A main challenge is the objective evaluation of functional and extra functional project goals to enable the assessment of the current development status. Even though quality modeling techniques to formalize quality characteristics exist, they did not reach a broader acceptance in practice so far. Companies often develop special solutions that are not reusable in other projects. This makes the quality measurement process itself less efficient and produces too much overhead.**

The method proposed in this paper is a contribution to solve these issues. It is applied at a German design house for analog mixed-signal IC design. This paper presents the results of a case study and introduces an optimized project scheduling on the basis of quality assessment results.

***Index Terms*— product development, project control, project scheduling, quality assessment**

I. Introduction

The studies [1], [2] (German) or [3] (English) show that it is still a big challenge to plan, manage and control development projects in almost every engineering discipline. Many projects exceed their deadline or budget or even fail completely due to poor project management.

The continuous, quantitative evaluation and tracing of the quality of products and intermediate results provide a basis for risk minimization and lead to an early detection of problems during development. For risk reduction it is necessary to evaluate whether quality goals can be fulfilled in time or not prior to reaching a milestone or gate deadline. Only then the project plan can be adapted and the priority of tasks adjusted in time. Furthermore, subjective and time intensive review processes should be more automated.

Beyond the general ISO 9000 definition of quality as a level of requirement fulfillment already Garvin [4] described quality as a complex concept with multiple facets and defined five different perspectives of quality (transcendent view, product view, user view, manufacturing view, value-based view).

Two of these perspectives are of particular interest for operative project control. Firstly, there is the product view that defines quality by specifying the quality relevant product characteristics. The second perspective is the manufacturing view that defines quality as conformance to the specification. The latter corresponds to several more definitions, as for example by Crosby [5] (“Conformance to requirements”), Gilmore [6] (“Quality is the degree to which a specific product conforms to a design or specification”). This shows that quality is a concept with multiple facets on the one hand and unique to each company and project on the other. Therefore, quality evaluation must be done in a goal oriented way to support the different perspectives. This makes it very hard to evaluate and even understand.

Therefore, this paper proposes a project control and planning system that comprises the following major two functionalities:

- A goal oriented quality assessment methodology that combines reusable quality models with existing requirements traceability techniques to adapt them to company and project specific needs.
- A method to efficiently plan design projects by means of project schedule optimization and re-scheduling based on quality assessment results.

The remainder of this paper is structured as follows: Section II presents related work. Section III introduces the developed project control and planning environment where both the quality assessment and the scheduling component are detailed. Section IV describes the application of the method on an analog mixed-signal design project use case. The paper closes with the conclusions in section V.

II. Related Work

A. Quality Assessment

Related work exists in different research areas and domains. Quite some work was published on the definition and measurement of software quality ([7], [7], [8], [9], [10]). In addition, there is the ISO standard [11] that defines a framework for software product quality. This standard uses similar hierarchical techniques to model quality as described in [7]. Especially the ISO standard is an example for the

Manuscript received July 18, 2009. This work was supported in part by the EU funded research project CESAR (ARTEMIS JU).

Stefan Häusler is with the R&D Division Transportation in the research institute OFFIS, Oldenburg, Germany (phone: +494419722582; fax: +494419722278; e-mail: haeusler@offis.de).

Axel Hahn is the professor of Business Engineering Division at University of Oldenburg and member of the board at the R&D Division Transportation at OFFIS (e-mail: hahn@offis.de).

Jana Blaschke and Christian Sebeke are with the Robert Bosch GmbH, Tübinger Straße 123, 72762 Reutlingen (e-mail: {jana.blaschke | Christian.sebeke}@de.bosch.com).

Wolfgang Rosenstiel is Chairman for technical computer science at the University of Tübingen (e-mail: rosentstiel@informatik.uni-tuebingen.de).

approach to define abstract quality criteria and refine them into measurable parameters.

Another tool for software quality evaluation is described in [12]. Quality models are created and configured based on the requirements of the specific project. Then, the quality is evaluated against product characteristics. One major drawback is the manual definition target values for every measurable parameter.

The authors in [13] mention the necessity for real-time quality evaluation to identify problems early in the development process of software products. The result of this research is a tool called ConQAT. The same authors propose an integrated approach to evaluate software quality ([14] and [15]). Only an integrated view on different quality criteria will help to reveal dependencies and inconsistencies. While their requirements and assumptions are very similar to those of this work, the carrying out is quite different. The tool mainly integrates several code analyzers. A meta-model for data integration is explicitly not used. The tools are directly connected. This solution seems sufficient for software development, but chip design requires many more tools and data sources. Therefore, it seems to be unlikely to integrate all these sources directly.

Related work in the domain of integrated circuits is quite rare and limited to quality evaluation of Intellectual Property (IP) components. In [16] quality metrics to evaluate IP's and System-on-Chip (SoC) were developed. Similar research was done in the project IPQ [17]. Nevertheless, this work is not suitable to evaluate the quality of non IP components. Furthermore, the evaluation is based on excel-sheet questionnaires and is not automated.

B. *Design Project Scheduling*

Scheduling of analog mixed-signal application specific integrated circuits (ASICs) design projects has been object to research for a long time [18]. But due to frequently changing design automation tools and new technologies ASIC design project planning requirements are constantly changing. Old approaches become obsolete. Recent research in design project scheduling focuses on selected design steps, as for example the high level synthesis of netlists [19], instead of scheduling the whole design project. Other models are based on derived metrics [20] and are found to be static, generalizing and take only a small portion of all influencing factors into account. Other approaches in the academic sector ignore a lot of influential factors such as used tools and libraries in their analysis [21][22]. Their analysis results do not allow the comparison of different designs or an individual planning for new projects.

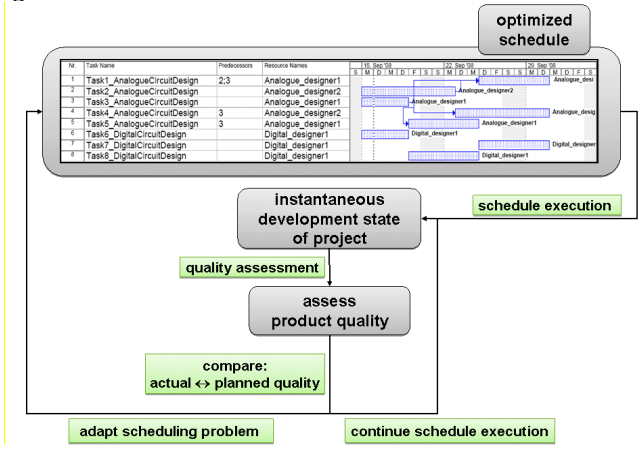

Fig. 1: Interaction between schedule optimization, project execution, quality calculation and re-scheduling

The scheduling of projects in general is a well known problem. First approaches introduced the critical path method and the metra potential method [23] to schedule a set of project activities. With the introduction of resource constraints the resource-constrained scheduling problem (RCPSP) [24] was obtained as a generalization of the project scheduling problem. The multi-mode RCPSP (MRCPSP) is an extension of the RCPSP. Solutions based on genetic algorithms (GA) were proposed by Özdamar [25] and Hartmann [26]. The application of a GA to design process scheduling introduces another dimension as different types of tasks (analog and digital frontend and backend) with different modes have to be considered.

III. Project Control and Planning Environment

In order to achieve an efficient execution of ASIC design projects, the project has to be planned carefully. This includes task scheduling at the start of a project, as well as re-planning during project execution. The project team has to reflect: can we hold defined milestones or is the current quality not high enough? A special challenge is the calculation of the deviation between original schedule and actual project course and the definition of a decision function when to trigger a re-schedule. A basis for these two functions forms the evaluation of the current design quality. In the next two sections we describe a method to evaluate and trace the design quality, as well as the schedule and re-schedule mechanisms for which the quality assessment results are one major input for. Fig. 1 shows the interactions between these modules.

A. *Real Time Quality Assessment*

A practical and efficient quality assessment faces a number of challenges:

Firstly, there are a lot of verification steps and quality assurance activities in every domain, but they often only look at a certain aspect. A statement about the integrated quality of a product, a component or a milestone fulfillment degree can only be made under consideration of all of these (functional and extra-functional) facets. In [18] multiple quality criteria for hardware components are mentioned. Another panel session at a design automation conference [28] discussed the necessity of an integrated interpretation of hardware verification results.

Secondly, quality is based on the fulfillment of defined requirements/goals for a product. Therefore, the adaptation of quality models to company and project goals is another challenge. This is formulated in a research agenda on software quality [29], but can be transferred to other engineering disciplines, too.

Thirdly, a product is usually divided into different components each with different intermediate results. For a well directed support of project control a clear definition of quality goals is needed that allows the quality evaluation of single components, their intermediate results, as well as for the complete product. For this reason, an explicit binding of goals to components and their intermediate results is needed to enable a quality evaluation at different granularities. This

challenge was mentioned in [13], too.

To address these challenges the methodology is based on a combination of quality modeling and existing requirements traceability approaches. This becomes manifested in Fig. 2 with a conceptual goal model and a quality meta-model. A common semantic basis is needed to treat quality characteristics in a uniform way. Every quality model/metric can be described by the elements of this quality meta-model based on defined domain models. Exemplarily, three domain models ('System Architecture', 'Physical Design' and 'Verification') are shown in Fig. 2. We use common techniques from software quality modeling to hierarchically decompose complex quality indicators into measurable parameters (see e.g. [7]). The dependency between indicators and measurable parameters, as well as the aggregation functions, can be described using the meta-model. In doing so, a quality model becomes available and can be connected as a goal to a specific component.

A goal is defined by several parameters: a minimum and a maximum target value, as well as tolerance limits that must not be exceeded. This allows the representation of hard and soft boundaries. In combination with an actual value, these parameters form a function that calculates a fulfillment degree for the defined goals. The actual value is calculated through the quality model/metric that is linked to the goal (based on its concrete type like e.g. functional goal or area constraint). After the actual value is calculated, it can be matched to the defined target to assess the goal fulfillment degree.

For each goal, it must be defined which component must fulfill the goal and when it should be fulfilled. There are project goals for intermediate results (requirements, design, implementation, verification), as well as for the final product. Therefore, relations are defined to connect goals to

- a complete product structure or one of its components (in Fig. 2 'System Architecture' and 'Physical Design' are two types of product structures)
- a milestone or a quality gate in the project model. Preliminary work [30] is used for the representation of project structures.

Once we have defined the quality plan, the current development status must be captured to perform a quality evaluation. For this purpose, the performance measurement framework Permeter, introduced in [31], is used. Permeter offers the functionality to load data from different sources into one integrated format, as well as manual and automatic

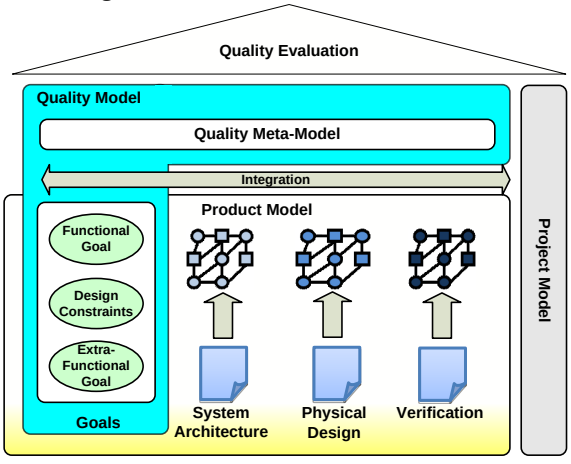

Fig. 2: Real-time quality assessment concept

linkage facilities to perform the data integration. The data is loaded via specific transformations into the used ontology format OWL (Web Ontology Language). For a detailed description of the data integration process see [31]. A frequent collection of the needed data documents captures the progress over time and makes it analyzable.

In general, this concept allows the measurement of the following values:

- milestone and quality gate fulfillment
- quality of single components
- quality of the complete product
- quality of intermediate results of the product

For a decent representation, the aggregated results are represented in reports. This will offer the possibility to quickly locate quality problems and the possibility to search for their possible causes.

B. Schedule Optimization and Rescheduling

The sensible planning of activities and procedures is an important aspect for the efficient organization and execution of projects. In a first step, the complexity of all involved tasks needs to be determined. The quality and fulfillment degrees of components (section III.A) form a good measurement for the task complexity. In a second step, these tasks need to be scheduled in a sophisticated manner. The scheduling needs to be carried out with respect to an optimal task arrangement, as well as an effective resource allocation. Furthermore the resulting schedule needs to consider many constraints regarding tasks and resources:

- dependencies between design tasks → precedence constraints
- tasks and designers belonging to different design regions → assignment constraints
- designers possessing different levels of design experience (modes) → modulation of task execution times
- designer availability

To receive an efficient project schedule, considering all these constraints, we developed a methodology based on a GA optimization [32]. The GA individuals possess two chromosomes that are to be optimized. The first chromosome λ optimizes the execution order of design tasks under consideration of task dependencies. The second chromosome μ optimizes the assignment of resources to tasks with respect to their constraints and properties. Fig. 3 exemplarily shows the two chromosomes of a GA individual, e.g. Task1 is scheduled in the thirteenth place, after Task2 and Task3 on which it is dependent. This example has three different execution modes, which modify the execution time of the tasks. The execution mode of Task1 is mode 1. The genome of the individuals is mapped to a valid schedule by a Schedule Generation Scheme (SGS). The SGS takes into account designer constraints and priorities of design tasks. Tasks with high priorities are scheduled earlier by the algorithm than tasks with low priority. These priorities can be defined according to fulfillment degrees of design components for example. Fig. 3 shows an example for a small ASIC design project. It depicts the translation of a GA genome into a valid project execution schedule by an SGS under consideration of a priority array. The available resources for that problem are one designer with mode 1 and one with mode 3 for analog circuit design knowledge and two

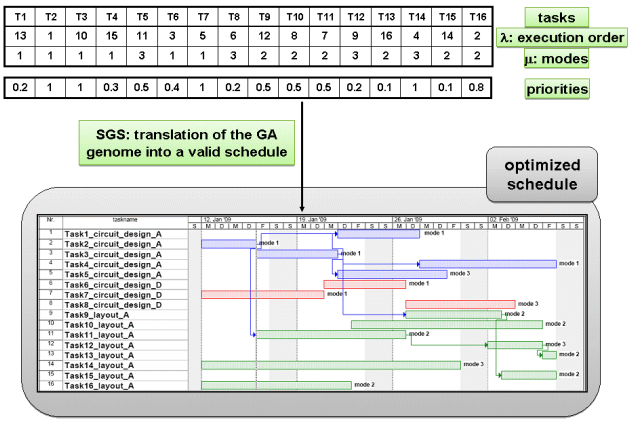

Fig. 3: Translation of a GA genome and an additional priority array into a valid project execution schedule by a Schedule Generation Scheme.

design tools. The same numbers apply for the digital designers, but with only one design tool available. For the layout design phase there are two analogue designers with mode 2 and one with mode3 and three tools available. The analogue designer with mode 3 is on vacation for the first six days of the project, the digital designer with mode 3 for the first seven days.

At the start of a project, an optimized schedule is produced by the GA as a task processing guideline. During project execution some project conditions may change though. Therefore, the project course can differ from the initial schedule. The schedule has to be compared to the project course, and if necessary to be adapted to the changed project conditions to guarantee an efficient performance with maximized resource utilization. The comparison is done in frequent time intervals.

The schedule-project course comparison, as well as the decision function when a re-schedule must be performed are based on the quality goals defined in section III.*A* and their degree of fulfillment at some point of the project execution. The schedule-project execution deviation at project execution time e_j is calculated as follows:

$$dev = \sum_{i=1}^{n} (\delta_{t_i}(pQ_j, aQ_j) \cdot (1 + a(t_i)) \cdot 0.01$$

n indicates the number of tasks and $a($ $)$ weights the number of dependent tasks to $t_i \cdot \delta_{t_i}(pQ_j, aQ_j)$ determines the difference between the planned and the actual quality of task t_i at execution time e_j by taking the difference of the planned and the measured quality value.

The decision function for a re-schedule is stated as followed:

$$re-sched(x) = \begin{cases} true, & x > 2.5 \\ false, & x <= 2.5 \end{cases}$$

By this means the execution schedule is connected with the quality plan.

If a re-schedule is triggered the GA optimization problem is adapted to the new project conditions. The following changes have to be considered:

- remove all finished tasks
- tasks that are being processed when a re-schedule is triggered are generating new resource constraints to be regarded in the problem
- the priority of tasks has to be adapted to the degree of quality fulfillment of their components

A new schedule that guarantees an efficient proceeding of the project is produced.

IV. Analog Mixed Signal Design Project Example

We evaluate the developed approach with design data of a real analog mixed-signal design project. We started to record the data some weeks after the project started. The data was collected from five different sources:

- specification → design requirements and goals
- excel sheet → definition of electronic devices
- VSDE (Virtuoso Specification-driven Environment) → verification data of analogue components
- layout meta- and constraints data
- project tracing

The data from the specification and excel sheet were used to capture the involved electronic components and their design goals. As the data of the design process changes constantly, we took screenshots of the product related data once in a week for seven weeks.

For each screenshot it is possible to evaluate the defined design goals against the development status. In the current version it is possible to formulate and evaluate area goals, analog quality goals and layout quality goals. We developed two quality models for the latter two types. The first model is defined to measure the quality of the analogue frontend and the second model describes how to measure the quality of the layout representation of a component (both analogue and digital parts). The hierarchical structure of both models is shown in Fig. 4below.

The analogue frontend quality model has the following three sub-indicators:

Criticality: This indicator is an index between 0 and 1 where 1 means that there are tests with results extremely close to the specification. 0 means that there is no critical test. The farther the results are inside the specification borders the better. Depending on the lowest criticality value (#minCPK) of a test and the number of total critical tests (#critCPK) a value is asserted to the 'criticality' indicator according to the following equation:

$$c = \begin{cases} 1, & \#minCPK < 1.4 \\ 0.5, & 1.1 <= \#minCPK < 1.4 \text{ and } \#critCPK < 5 \\ 0, & \#minCPK < 1.1 \text{ or } \#critCPK \geq 5 \end{cases}$$

A test is assumed to be critical with a CPK value < 1.4. The CPK value is adopted from manufacturing and is calculated for min and max specification boundary as shown in the equation below. The lower value of the two is set to minCPK attribute of the test.

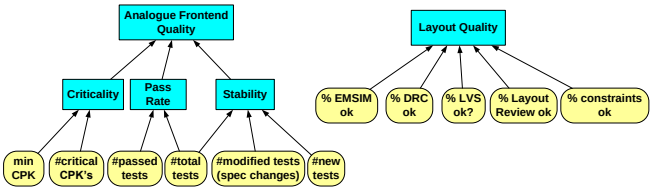

Fig. 4: (left) analogue frontend quality model, (right) layout quality model

$$\text{maxCPK} = \frac{\text{maxSpec} - 0.5(\text{maxVal ue} + \text{minValue})}{0.5(\max Value - \min Value)}$$

$$\text{maxCPK} = \frac{0.5(\text{maxVal ue} + \text{minValue}) - \text{minSpec}}{0.5(\max Value - \min Value)}$$

Pass Rate: This indicator is an index between 0 and 1 that shows how many tests pass successfully according to its specification. The value is simply calculated using the following equation:

$$\text{PassRate} = \frac{\#\text{passedTests}}{\#\text{totalTests}}$$

Stability: The last indicator gives information about the current stability of all tests. This indicator was introduced under the assumption that high volatile test definitions are not of high quality. Both the rate of new measures compared to total measures and the rate of modified measures (spec changes) are considered in the stability calculation:

$$stability = 1 - \frac{\#\text{newMeasures} + \#\text{modifiedMeasures}}{\#\text{totalMeasures}}$$

The analogue quality value is calculated by the average of the three mentioned indicators. The introduction of optional indicator weights is possible.

The layout quality for a component is calculated as the average of the following parameters:

- percentage of subcomponents with robustness check ok
- percentage of subcomponents with logic vs. schematic (LVS) ok
- percentage of subcomponents with design rule check (DRC) ok
- percentage of subcomponents with layout review ok
- percentage of subcomponents with constraints ok

Once formalized and added to the quality model repository of the implemented tool Permeter, these models become available to link them as goals to a component and to a milestone. To do so, we have to define the domain models as described in section III. There are two domain models describing the design structure at different abstraction levels. Analog/digital backend describes the AMS design in its layout representation; system architecture is an early abstract hierarchy of the design. Analog frontend represents all verification results for analog components. These models including the relations to link them are illustrated in Fig. 5.

For each of the domain models a transformation is defined that loads the current development status from its propriety data source into the OWL format defined by the domain model. Then, the elements are linked to each other using manual and (semi-) automatic linkage facilities available in Permeter to establish the needed traceability. The result is an integrated model of formally distributed data allowing for an integrated quality evaluation based on the defined goals and their underlying quality models. A goal is evaluated against the component it is linked to. To perform the goal definition step, Permeter offers a perspective where goals can be created and linked to a milestone of a loaded project plan (e.g. from MS-Project) or to a component of one of the product structures it can be linked to. A project cockpit shows the quality trend over the seven weeks of development we tracked. You can drill down into each snapshot and navigate through the defined system architecture to see quality values on component level. Furthermore, you can drill into the goals and into its quality model values itself to locate possible quality problems. For the project plan milestone fulfillment degrees are shown.

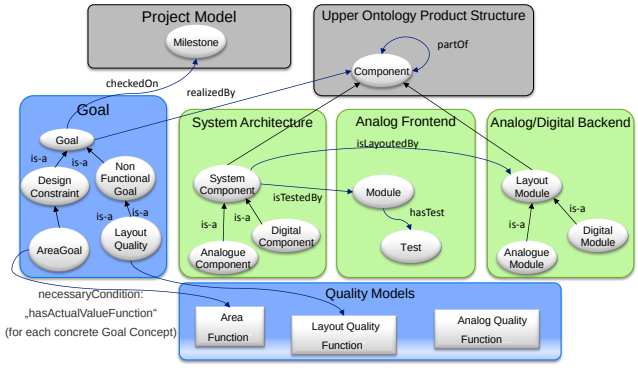

Fig. 5: Goals can be linked to components and to milestones. Each goal type has one associated quality model/metric defined as described in this section.

The calculated quality measures are one input to calculate an optimized project schedule. The input design tasks for the project schedule are derived from the project tracing source and the quality fulfillment of the components at week 1 (the project was already in progress when we started to track design data). Based on this data we generated an optimized schedule with the GA according the task execution order and task-resource assignment as shown in Fig. 6. The planned quality pq_j^x of every component x at time instance j was calculated according to its completion degree after every scheduled week. If a component was finished to 60 % after termination of week one, the planned quality was also set to 60 %. Then, we determined the overall quality of the whole project by calculating the mean of all components:

$$pQ_j = \frac{1}{n}\sum_{x=1}^{n} pq_j^x .$$

The actual quality aq_j^x for every component x was calculated according to their type, analog frontend or layout task, as described above. The actual quality of the whole project was determined in equal measure to the overall planned quality. For each week, we calculated the deviation between planned and measured quality (see top of Fig. 6). For this we compared the deviation for every single task and calculated the mean for the overall quality. During the first

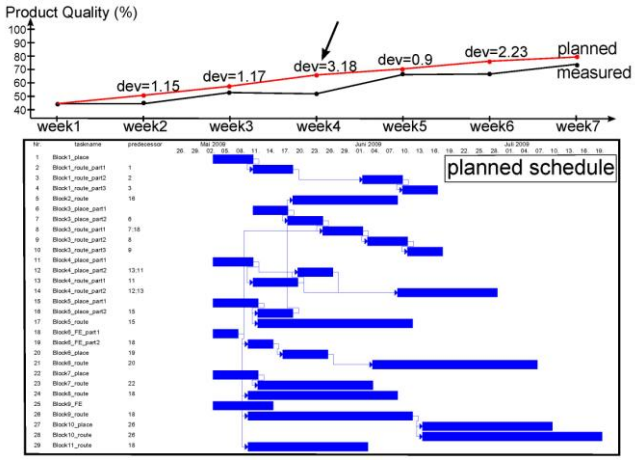

Fig. 6: Optimized schedule for the remaining design tasks of the project. The according planned quality and the measured quality are shown together with the schedule-execution deviation in the graph above.

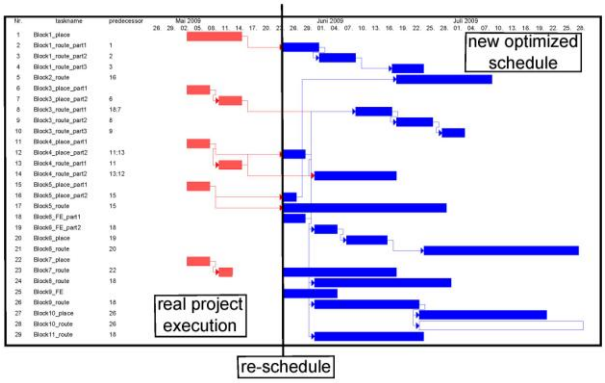

Fig. 7: The first part of the figure shows the executed design tasks of the first three weeks of the project, painted in red. The second part shows the optimized schedule of the remaining tasks.

two weeks, the deviation is below the defined threshold of 2.5. A re-schedule is not triggered. This changed in the third week, when the deviation reached a value of 3.18. As the decision function triggers a re-schedule at a deviation of 2.5, a new schedule is generated in week three. Fig. 7 depicts the processed tasks during the first three weeks of the project and the new optimized schedule obtained after the re-schedule.

As the project team did not follow the optimized schedule, the project did not proceed as fast as it could have. Accordingly, the product quality did not increase as much as it was planned. After three weeks of project execution it differed as much from the schedule that a re-schedule was triggered. If the project would be continued according to the old schedule by that time, this would result in an inefficient remaining project course. The new schedule allows an efficient organization for the remaining project part.

V. Conclusions and Future Work

We developed a system that allows tracking and optimization of chip design projects. By defining and monitoring the product quality of the chip we get a measure for the degree of completion for a project and individual design tasks. Based on this, we can generate a schedule that optimizes the task execution order and task-resource assignments. Tracing and comparing the planned and the actual quality of components, the system allows the initiation of a re-schedule if necessary. By this way a sensible and efficient project execution is guaranteed for the whole project. Additional quality models can easily be integrated into the system so it is usable for other domains and companies as well.

References

[1] U. Harmuth, “Erfolgsfaktoren für Projekte, Analyse von PM-Award-Projekten nach gemeinsamen Erfolgsfaktoren”, 2003.
[2] R. Buschermoehle, H. Eekhhoff, and B. Josko, ”SUCCESS rate and factors of IT-Projects 2006 in Germany”, In Proc. of the International Conference on Software Engineering Research and Practice, Las Vegas, pp.25-28, 2007.
[3] The Standish Group, The CHAOS Report, 1994.
[4] D. A. Garvin, “What does product quality really mean?”, MIT Sloan Management Review, vol. 26(1), pp. 25–43, 1984.
[5] P.B.Crosby, “Quality is free”, Reissue Edition, Signet, 1980.
[6] H.L.Gilmore, “Product Conformance Cost”, Quality Progress, pp.16-19, 1974.
[7] B.W. Boehm, J.R. Brown, M. Lipow, G.J. MacLeod, M.J. Merritt, “Characteristics of SoftT.P. Bowen, “Specification of Software Quality Attributes”, Rome Laboratory, New York, Technical Report, RADC-TR-85-37, Vols. 1-3, 1976.
[8] M. Deutsch and R. Willis, “Software Quality Engineering”, Prentice-Hall, 1988 Quality Engineering”, Prentice-Hall, 1988.
[9] B. Kitchenham, “Towards a Constructive Quality Model”, Soft. Eng. Journal, 105-112, July 1987.
[10] R. Nance, “Software Quality Indicators: An Holistic Approach to Measurement”, Proc. 4th Annual Software Quality Workshop, Alexandria Bay, New York, August 1992.
[11] ISO/IEC 9126, “Information technology -- Software product evaluation -- Quality characteristics and guidelines for their use”, Beuth-Verlag, Berlin, 1991.
[12] J. Boegh, S. Depanfilis, B. Kitchenham, and A. Pasquini, “A method for software quality planning, control, and evaluation”, IEEE Software, 16(2), pp. 69 – 77, 1999.
[13] F. Deissenboeck, E. Juergens, B. Hummel, S. Wagner, B. Mas y Parareda, and M. Pizka, “Tool Support for Continuous Quality Control”, IEEE Software, vol. 25, no.5, p. 60-67, 2008.
[14] M. Broy, F. Deißenböck, and M. Pizka, “A holistic approach to software quality at work.” In 3rd World Conference for Software Quality (3WCSQ), Munich, Germany, September 2005.
[15] S. Wagner and F. Deissenböck, “An Integrated Approach to Quality Modelling”. In Proceedings of the 5th Workshop on Software Quality, 2007.
[16] M. Birnbaum and C. Johnson, “VSIA Quality Metrics for IP and SoC”, International Symposium on Quality Electronic Design, pp. 279-283, San Jose, CA, USA, 2001.
[17] H.J. Brand, S. Rülke, and M. Radetzki, “IP Qualification for Efficient System Design“. Proceedings of the 5th International Symposium on Quality Electronic Design (ISQED), 2004.
[18] S.J. Easely and E.E. Freeman, “ASIC Design Project Scheduling”, *ASIC Seminar and Exhibit, 1989. Proceedings., Second Annual IEEE*, P11-2/1-4, 1989.
[19] Chi-Ho Lin, Jin-Chun Kim, “A Novel Scheduling methodology for ASIC Design”, *System-on-Chip for Real-Time Applications, The 6th International Workshop on*, 131-134, 2006.
[20] P. Leppelt, A. Hassine, and E. Barke, “An Approach to Make Semiconductor Design Projects Comparable”, *7th Asia Pacific Industrial Engineering and Management Systems Conference*, 2006.
[21] E. Johnson, “Analysis and Refinement of Iterative Processes”, Doctoral Thesis , University of Notre Dame, Notre Dame, IN: 1996.
[22] P. Sutton, “A Framework and Discipline Independent Approach to Design Process Management”, Doctoral Thesis, Carnegie Mellon University, Pittsburgh, PA: 1997.
[23] Neumann/Morlock, “*Operations Research : mit 111 Tabellen* ”, Hanser Verlag München Wien, 1993.
[24] J. Blazewicz, J. K. Lenstra, and A. H. G. R. Kan, "Scheduling subject to resource constraints: Classification and complexity," *Discrete Applied Mathematics*, vol. 5, pp. 11-24, 1983.
[25] L. Özdamar, “A genetic algorithm approach to a general category project scheduling problem”, *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 29, pp.44-59, 1999.
[26] S.Hartmann, “Project scheduling with multiple modes: a genetic algorithm”, *Annals of Operations Research*, 102, pp. 111—135, 1997.
[27] A.H. Farrahi, D.I. Hathaway, M. Wang, and M. Sarrafiadeh, “Quality of EDA CAD Tools: Definitions, Metrics and Directions”, In Proc. IEEE Intl. Symp. On Quality Electronic Design, pp. 395-405, March 2000.
[28] F. Bacchini, A.J. Hu, T. Fitzpatrick, R. Ranjan, D. Lacey, M. Tan, A. Piziali, and A. Ziv, “Verification coverage: when is enough enough”. Proceedings of the 44th annual conference on Design automation, San Diego, California, 2007.
[29] S. Wagner, M. Broy, F. Deissenboeck, M. Kläs, P. Liggesmeyer, J.Münch, and J.Streit, “Eine Forschungsagenda für Softwarequalität”, *In Workshop-Band Software-Qualitätsmodellierung und -bewertung (SQMB '08)*, Technical University Munich, 2008.
[30] A. Hahn, S. grosse Austing, J. Strickmann, “Metrics – The Business Intelligence Side of PLM”, Proceedings of the International Conference on Product Lifecycle Management, pp. 11 – 20, 2007.
[31] A. Hahn, K. Hausmann, S. Häusler, and J. Strickmann, “Using Ontologies to Model and Understand Product Development“. In IBIS – Interoperability in Business Information Systems, 2(5), pp.21-39, 2007.
[32] J. Blaschke, C. Sebeke, and W. Rosenstiel, “Using Genetic Algorithms for Planning of ASIC Chip-Design Project Flows”, IEEE Congress on Evolutionary Computation, pp. 1881-1888, June, 2009.