Using COBIT Indicators for Measuring Scrum-based Software Development

VILJAN MAHNIC, NATASA ZABKAR
Faculty of Computer and Information Science
University of Ljubljana
Trzaska 25, SI-1000 Ljubljana
SLOVENIA
viljan.mahnic@fri.uni-lj.si, nzabkar@email.si, http://ltpo.fri.uni-lj.si/

Abstract: - The aim of this paper is to determine the level of compliance of AGIT model, developed during our previous research for measuring Scrum-based software development, with the information systems auditing criteria. For this purpose we use COBIT model. After a short introduction of Scrum, AGIT and COBIT, we perform comparison analysis of their indicators for software development. Then we upgrade AGIT model with the selected COBIT indicators. In order to improve the clarity of the model, we present its structure using IT Balanced Scorecard. Finally we suggest possible further research.

Key-Words: - Scrum, Agile software development, IT performance measurement, IT indicators, IT Balanced Scorecard, COBIT, AGIT

1 Introduction

Software development projects are often of key importance for the achievement of the organisation's mission and objectives in the effective and efficient, transparent and auditable manner [7]. There have been many attempts to reduce the risks of software defects and improve the quality of software development process. These attempts include introduction of agile methods and recently introduction of project management mutation model, which incorporates agile methods and all other methods developed so far [20].

Introduction of agile methods can change and improve project management practices [3], decrease overtime and increase customer satisfaction [19]. The success rate is 41% for the agile projects and 16% for the waterfall projects, according to [31], who refers to The Standish Group 2006 research report.

XP and Scrum are the most commonly used agile methods [24]. In the last few years several successful implementations of Scrum have been reported in the literature ([22], [30], [28], [2]). According to [28], the usage of Scrum reduces every category of work (defects, rework, total work required, and process overhead) by almost 50%, when used in CMMI level 5 compliant company.

In this paper we focus on Scrum performance measurement. Like many other agile software development methods, Scrum follows the principle of "maximizing the amount of work that need not be done". Therefore, it abandons many practices prescribed by software quality models including the need for comprehensive metrics plans. Many authors have recognized and explored a need for more elaborate metrics for agile development ([1], [27], [4]).

Our previous research in this area is summarized in the AGIT (AGIle software developmenT) model, which includes basic indicators for measurement of Scrum-based software development [15], the introduction of CMMI Measurement and Analysis Practices into Scrum-based Software Development Process [17] and a description of corresponding measurement repository [16]. In this paper we further explore indicators of AGIT model. Our intention is to determine the level of compliance with the information systems auditing criteria. For this purpose we use COBIT (Control Objectives for Information and Related Technology) [8], the IT governance framework that is generally accepted in the information systems auditing community and commonly used for IT governance implementation and assessment. COBIT has also been used from the auditing perspective of agile development in [5], for determining compliance of the projects using agile techniques with Sarbanes Oxley Act (SOX), a regulatory requirement for all public listed companies in United States.

After a short introduction of Scrum, AGIT and COBIT model, we describe COBIT indicators for system development life cycle. Then we explain our compliance criteria. After that we compare Scrum, AGIT and COBIT indicators and discuss the results of this comparison. This is followed by the proposal of the adjustments to our model, including new presentation of its structure. In the end, we give conclusions and plans for future work.

2 Scrum

2.1 Introduction to Scrum

In this paper we assume that the reader is already familiar with Scrum-based software development process [23].

One of the key Scrum tools for the performance measurement is a direct day-to-day monitoring at the 15-minute Daily Scrum meeting. At this meeting every Team member answers three questions:

- What have you done on this project since the last Daily Scrum meeting?
- What will you do before the next meeting?
- Do you have any obstacles?

ScrumMaster is responsible for these meetings and for resolving impediments encountered during the Sprint in order to assure smooth running of the development process. During these meetings data for the performance measurement can be gathered.

2.2 Scrum Indicators

Within Scrum originally only one software development metric is used. This original Scrum indicator is the estimate of the amount of work remaining that needs to be done in order to complete a Product Backlog item or a task in a Sprint Backlog.

The total amount of work remaining is usually shown in the burndown chart [23]. An example of the burndown chart is presented in Fig. 1, based on the data from [13].

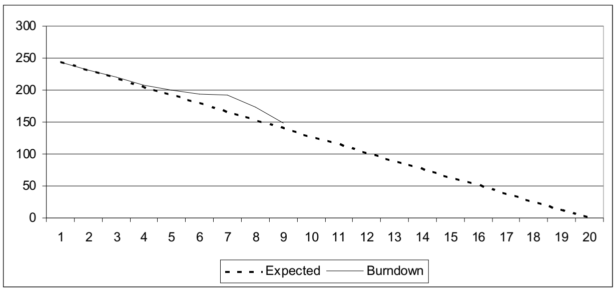


Fig. 1: Sprint burndown Chart

We can see that Scrum Team was on the schedule until the 5th day, when they started going behind the schedule (the “burndown” line is above “expected” line), but was back on the course at the 9th day.

Scrum teams often use the Scrum board for tracking [Sut07b, Gup08]. On Scrum boards they post Scrum burndown chart, prioritized list of impediments, columns of user stories, development tasks relating to each story and the state of each development task and tests associated with each story, which can be further visualized by lamps that change color depending on the state of the build or state of the Sprint. The usage of Scrum board eliminates the need for most status reporting, since managers can see the state of the team in a few seconds. This is even easier if critical information is put online on a web page, a wiki, or a reporting tool.

According to [1], Scrum provides four simple and effective artifacts for managing and monitoring project performance:

- Product Backlog,
- Product Burndown,
- Sprint Backlog and
- Sprint Burndown.

The progress can be presented in a general view like a burndown, which is easy to follow and understand by managers, and a detail level like a backlog, which is useful for tracking the results.

In CMMI level 5 compliant companies [10], Scrum progress (of sprints) is primarily measured through the sprint burn down chart and the sprint review meeting. [28] suggests combination of Scrum Burndown Chart and defect data, since Scrum compensates other indicators by its high level of interaction at the meetings

The indicators stated above are based on the original Scrum indicator: estimate of the amount of work remaining. In the next section we introduce additional indicators, without violating the principles of agility.

3 AGIT Model

3.1 Introduction to AGIT Model

Experience has shown ([13], [14]) that, beside estimating the amount of work remaining, it is useful to measure the information about the amount of work spent. This can be easily collected during daily Scrum meetings and used for calculating the Earned Value [21].

Based on this experience, we have developed AGIT (AGIle software development) model. According to the principles of stakeholder driven process performance measurement [12], the best performance is achieved when the goals of all stakeholders are satisfied. This requires a balanced approach considering viewpoints of different stakeholders, so AGIT model describes the appropriate indicators for each stakeholder [15].

In AGIT model the views of four different stakeholders are considered:

- IT management,
- Team members,
- ScrumMaster and
- Customers.

The first stakeholder is IT management who is mainly concerned with traditional aspects of software development performance considering time, cost, and quality. The second stakeholder is Team members whose main goal is “Job satisfaction”. The ScrumMaster is the third stakeholder with the main goal of “Efficient impediments resolution”. Finally, the main goal regarding the customers, the fourth stakeholder, is “Customer Satisfaction”.

The top-down approach has been used in AGIT model, in order to define the goals of each stakeholder, appropriate performance indicators and metrics that enable the evaluation of each indicator, taking into account that indicators should describe the process quantitatively and qualitatively [15]. AGIT also includes a generic data model of measurement repository for collecting and storing measurement results [16].

The important care was taken not to violate the principles of agility. All metrics (except the number of errors reported by the user after release) can be collected during meetings already prescribed by Scrum, thus not requiring a substantial additional effort of the Team.

3.2 AGIT Indicators

The AGIT indicators for the previously stated goals are shown in Table 1. Detailed description and formulas of these 12 indicators can be found in ([15], [17]).

Table 1: Indicators for Scrum-based Software Development Process (AGIT)

Stakeholder	Goal	AGIT Indicator
AG1: IT Management	Timely Information on Project Performance	**AG1-1:** Work Effectiveness (ratio between the work spent and the decrement of work remaining)
		AG1-2: Schedule Performance Index (SPI) (ratio between the earned value (i.e., the value of all tasks completed) and the planned value (i.e., the initial estimate of value of all tasks to be completed till a certain point within the project))
		AG1-3: Cost Performance Index of Labor Costs (CPI) (ratio between the earned value (measured in units of currency) and actual costs)
	Quality Improvement	**AG1-4:** Error Density (number of errors per KLOC (kilo-lines of code))
		AG1-5: Costs of Rework (product of hours spent on rework and cost of an engineering hour)
		AG1-6: Fulfilment of Scope (ratio between the number of tasks completed in the Sprint and total number of tasks in the Sprint Backlog or between the number of PBIs completed in the release and total number of PBIs committed)
AG2: Team Members	Job Satisfaction	**AG2-1:** The Average Amount of Overtime at Sprint/Release/Project level (the expected hours, the amount of work spent and administrative days)
		AG2-2: The Average Number of Projects the Employees Work in Parallel
		AG2-3: Qualitative Evaluation of Working Conditions (communication and teamwork, physical discomfort, psychological well-being, workload, supervision, opportunities for growth, etc.)
AG3: ScrumMaster	Efficient Impediments Resolution	**AG3-1:** Average Number of Impediments per Task/Sprint/Team
		AG3-2: Mean Time for Resolving an Impediment (at Task/Sprint/Team level)
AG4: Customers	Customer Satisfaction	**AG4-1**: Qualitative Evaluation of Customer Satisfaction (the quality of product, price adequacy, reliability in terms of time and costs, completeness of product delivered at the end of each Sprint or release, flexible handling of changes in requirements, good collaboration with the development team, adequate training and documentation, etc) **AG1-4:** Error Density **AG1-6:** Fulfilment of Scope

4 COBIT Model

4.1 Introduction to COBIT Model

COBIT (Control Objectives for Information and Related Technology) [8] represents a collection of documents which can be classified as generally accepted best practice for IT governance, control and assurance. Our hypothesis is, that by satisfying information systems auditing criteria, we can demonstrate that Scrum indicators proposed by AGIT are (or are not) compliant with good practice.

The IT processes are usually ordered into the responsibility domains of plan, build, run and monitor. Within the COBIT framework, these domains are called:

- **Plan and Organise (PO):** Provides direction to solution delivery (AI) and service delivery (DS);
- **Acquire and Implement (AI):** Provides the solutions and passes them to be turned into services;
- **Deliver and Support (DS):** Receives the solutions and makes them usable for end users;
- **Monitor and Evaluate (ME):** Monitors all processes to ensure that the direction provided is followed.

Across these four domains, shown in Fig. 2, COBIT has identified 34 IT processes.

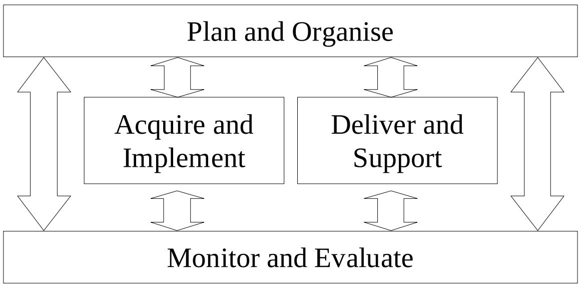


Fig. 2: COBIT Domains [8]

For each of its 34 processes COBIT defines goals and metrics to define and measure their outcome and performance, based on the principles of the balanced business scorecard (BSC), introduced by Robert Kaplan and David Norton [11]. COBIT metrics have been developed with the following characteristics in mind:

- A high insight-to-effort ratio;
- Comparable internally an externally;
- Better to have a few good metrics;
- Easy to measure.

These characteristics are mainly compliant with the agility principles that value individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan [18].

The COBIT indicators can be used as key performance indicators (KPI), key goal indicators (KGI) or key risk indicators (KRI). They can be integrated in different models, such as Corporate IT Risk Management model [25].

In this paper we assess the compliance of AGIT and COBIT indicators and explore whether non-compliant or partly-compliant COBIT indicators can be integrated in AGIT model.

4.2 COBIT Indicators for Software Development Process

In order to assess Scrum-based development process indicators, we need to select the indicators for those COBIT processes that relate to software development. There are few possible options ([5], [6], [7], [32]).

Our selection of COBIT processes (and its indicators) is based on:

- ISACA IS Auditing Guideline for System Development Life Cycle (SDLC) Reviews (Guideline G23) [6],
- four AI processes (AI1, AI2, AI6, AI7) selected by [5] and
- additional two processes, which we have considered to be relevant for the agile development (PO7, DS10).

The guideline G23 states that COBIT guidance for the following 5 processes should be considered relevant when performing the audit of the system development life cycle:

- Process **PO8:** “Manage Quality” is focused on ongoing performance monitoring against predefined objectives.
- Process **PO10:** “Manage Projects” is focused on monitoring of project risks and progress.
- Process **AI1:** “Identify Automated Solutions” is focused on identifying technically feasible and cost-effective solutions.
- Process **AI2:** “Acquire and Maintain Application Software” is focused on ensuring that there is a timely and cost-effective development process.
- Process **DS5:** “Ensure Systems Security” is focused on defining IT security policies,

plans and procedures, and monitoring, detecting, reporting and resolving security vulnerabilities and incidents.

According to [5], the following phases of system development life cycle can be mapped to agile development:

- Understanding Requirements,
- Designing Solutions,
- Building Solutions,
- Testing Solutions and
- Implementing Solutions.

These phases are mapped to four AI processes (AI1, AI2, AI6, AI7). The first two processes are already included in the G23 selection so we added the following two processes:

- Process **AI6:** "Manage Changes" is focused on controlling impact assessment, authorisation and implementation of all changes to the IT infrastructure, applications and technical solutions; minimising errors due to incomplete request specifications; and halting implementation of unauthorised changes.
- Process **AI7:** "Install and accredit Solutions and Changes" is focused on testing that applications and infrastructure solutions are fit for the intended purpose and free from errors, and planning releases to production.

Finally, we have slightly broadened this selection by adding two COBIT processes which are, in our opinion, important for achieving the goals of stakeholders Team Members (PO7) and ScrumMaster (DS10):

- Process **PO7:** "Manage Human Resources" is focused on hiring and training personnel, motivating through clear career paths, assigning roles that correspond with skills, establishing a defined review process, creating position descriptions and ensuring awareness of dependency on individuals.
- Process **DS10:** "Manage Problems" is focused on recording, tracking and resolving operational problems; investigating the root cause of all significant problems; and defining solutions for identified operations problems.

The 26 indicators, stated in the process description for these 9 COBIT processes, are presented in Table 2.

Table 2: COBIT Indicators for System Development Life Cycle

COBIT Process	COBIT Indicators
*Domain **PO***	
PO7: Manage Human Resources	**PO7-1**: Level of stakeholders' satisfaction with IT personnel expertise and skills
	PO7-2: IT personnel turnover
	PO7-3: Percent of IT personnel certified according to job needs
PO8: Manage Quality	**PO8-1:** Percent of stakeholders satisfied with IT quality (weighted by importance)
	PO8-2: Percent of IT processes that are formally reviewed by QA on a periodic basis and that meet target quality goals and objectives
	PO8-3: Percent of processes receiving QA review
PO10: Manage Projects	**PO10-1**: Percent of projects meeting stakeholders expectations (on time, on budget and meeting requirements—weighted by importance)
	PO10-2: Percent of projects receiving post-implementation reviews
	PO10-3: Percent of projects following project management standards and practices
*Domain **AI***	
AI1: Identify Automated Solutions	**AI1-1:** Number of projects where stated benefits were not achieved due to incorrect feasibility assumptions
	AI1-2: Percent of feasibility studies signed off by the business process owner
	AI1-3: Percent of users satisfied with functionality delivered
AI2: Acquire and Maintain Application Software	**AI2-1:** Number of production problems per application causing visible downtime
	Percent of users satisfied with the functionality delivered (**AI2-2=AI1-3)**
AI6: Manage changes	**AI6-1:** Number of disruptions or data errors caused by inaccurate specifications or incomplete impact assessment
	AI6-2: Amount of application or infrastructure rework caused by inadequate change specifications
	AI6-3: Percent of changes that follow formal change control processes

AI7: Install and Accredit Solutions and Changes	**AI7-1:** Amount of application downtime or number of data fixes caused by inadequate testing
	AI7-2: Percent of systems that meet expected benefits as measured by the post-implementation process
	AI7-3: Percent of projects with a documented and approved testing plan
Domain ***DS***	
DS5: Ensure Systems Security	**DS5-1**: Number of incidents damaging the organisation's reputation with the public
	DS5-2: Number of systems where security requirements are not met
	DS5-3: Number of violations in segregation of duties
DS10: Manage Problems	**DS10-1**: Number of recurring problems with an impact on the business
	DS10-2: Percent of problems resolved within the required time period
	DS10-3: Frequency of reports or updates to an ongoing problem, based on the problem severity

5 Compliance Assessment

5.1 Method of Comparison

The compliance assessment is performed by determining the level of compliance for each COBIT indicator, selected in the previous section.

We compare the amount of information provided by each COBIT indicator with the information provided by selected AGIT indicators or Scrum method. The possible results of comparison are:

- Compliant,
- Partly compliant and
- Non-compliant.

If the information is comparable, then the indicators are compliant. If the amounts of information are significantly different, but still comparable, we mark indicators as partly compliant. Finally, if the information needed for COBIT indicator cannot be provided through any of AGIT indicators, we put these COBIT indicators in the non-compliant category.

Since COBIT addresses higher level of software development process than AGIT, the comparison of these indicators is not always possible at the indicator level. This is in line with the results of Scrum and CMMI comparison ([28], [10]), where many of the generic practices were considered to be integrated in Scrum and therefore 100% compliant.

5.2 Comparison Results

The results of comparison are presented in tables for each domain (Table 3-5).

5.2.1 Domain Plan and Organise (PO)

The results for domain PO are shown in Table 3.

PO7-1: Level of stakeholders' satisfaction with IT personnel expertise and skills
Status: Compliant
AGIT indicator AG4-1: "Qualitative Evaluation of Customer Satisfaction using Criteria" is compliant with this COBIT indicator.

PO7-2: IT personnel turnover
Status: Partly compliant
This COBIT indicator is partly covered by the three AGIT indicators that relate to stakeholder "Team members ":

- AG2-1: The Average Amount of Overtime at Sprint/Release/Project level;
- AG2-2: The Average Number of Projects the Employees Work in Parallel;
- AG2-3: Qualitative Evaluation of Working Conditions.

PO7-3: Percent of IT personnel certified according to job needs
Status: Non-compliant
There is no AGIT indicator that would be compliant with this COBIT indicator.

PO8-1: Percent of stakeholders satisfied with IT quality (weighted by importance)
Status: Compliant
This COBIT indicator can be mapped to AGIT indicator AG4-1: "Qualitative Evaluation of Customer Satisfaction using Criteria". Apart from this, at the end of the Sprint, every stakeholder can assess product quality at a Sprint review meeting at which the Team presents what was developed during the Sprint to the Product Owner and any other stakeholders who want to attend.

PO8-2: Percent of IT processes that are formally reviewed by QA on a periodic basis and that meet target quality goals and objectives, **PO8-3:** Percent of processes receiving QA review
Status: Compliant
These two COBIT indicators are covered by the Sprint retrospective meeting. According to [10], Scrum teams can use a quality assurance schedule (QAS), where it is outlined what quality activities

will be used to ensure the quality objectives are achieved.

PO10-1: Percent of projects meeting stakeholder's expectations (on time, on budget and meeting requirements—weighted by importance)
Status: Compliant
This COBIT indicator can be mapped to the following three AGIT indicators:

- AG1-1: Work Effectiveness;
- AG1-2: Schedule Performance Index (SPI);
- AG1-3: Cost Performance Index of Labor Costs (CPI).

PO10-2: Percent of projects receiving post-implementation reviews
Status: Partly compliant
Since Scrum Retrospective meeting happens immediately after Scrum Review meeting and before the next Scrum Planning meeting, and the post-implementation review is scheduled at a reasonable time after the IT solution has been implemented (four weeks to six months) (G29, [6]), there could be different views about the level of compliancy. However, the primary goal of post-implementation review is to assess the effectiveness and efficiency of the IT solutions and their implementation, initiate actions to improve the solution (where necessary) and serve as a learning tool for the future. This is reached through Scrum Sprint Retrospective meeting, which supports the goal of learning across projects by collecting the results from individual projects [28] and is an opportunity for the team to discuss what's working and what's not working, and agree on changes to try [5]. Therefore, our final decision was that this COBIT indicator is partly compliant.

PO10-3: Percent of projects following project management standards and practices
Status: Compliant
This COBIT indicator is covered by the ScrumMaster role, under the assumption that Scrum is used as a project management standard.

Table 3: COBIT Indicators for System Development Life Cycle – Domain PO

COBIT Indicator	*Status*	AGIT Indicator/Scrum requirement
PO7-1: Level of stakeholders' satisfaction with IT personnel expertise and skills	*Compliant*	**AG4-1:** Qualitative evaluation of customer satisfaction
PO7-2: IT personnel turnover	***Partly compliant***	**AG2-1:** The average amount of overtime **AG2-2:** The average number of projects the employees work in parallel **AG2-3:** Qualitative evaluation of working conditions
PO7-3: Percent of IT personnel certified according to job needs	***Non-compliant***	
PO8-1: Percent of stakeholders satisfied with IT quality	*Compliant*	**AG4-1:** Qualitative evaluation of customer satisfaction *Scrum: Participation at the Sprint review meeting*
PO8-2: Percent of IT processes that are formally reviewed by QA on a periodic basis and that meet target quality goals and objectives	*Compliant*	*Scrum: Sprint retrospective meeting*
PO8-3: Percent of processes receiving QA review	*Compliant*	*Scrum: Sprint retrospective meeting*
PO10-1: Percent of projects meeting stakeholders expectations	*Compliant*	**AG1-1:** Work effectiveness **AG1-2:** Schedule Performance Index **AG1-3:** Cost Performance Index of Labor Costs
PO10-2: Percent of projects receiving post-implementation reviews	***Partly compliant***	*Scrum: Sprint retrospective meeting*
PO10-3: Percent of projects following project management standards and practices	*Compliant*	*Scrum: Covered by the ScrumMaster role assuring that Scrum is used as a project management standard*

5.2.2 Domain Acquire and Implement (AI)

The results for domain AI are shown in Table 4.

AI1-1: Number of projects where stated benefits were not achieved due to incorrect feasibility assumptions
Status: Partly compliant

This COBIT indicator is partly covered by the AGIT indicator AG1-6: "Fulfillment of Scope", under the assumption that the reasons for non-completion of the Sprint tasks are related to incorrect feasibility assumptions.

AI1-2: Percent of feasibility studies signed off on by the business process owner
Status: Compliant
This COBIT indicator is covered by the Sprint planning meeting.

AI1-3, AI2-2: Percent of users satisfied with functionality delivered
Status: Compliant
These two COBIT indicators as well as previously described COBIT indicator **PO8-1** (Percent of stakeholders satisfied with IT quality) can be mapped to AGIT indicator AG4-1: "Qualitative Evaluation of Customer Satisfaction using Criteria".

AI2-1: Number of production problems per application causing visible downtime
Status: Compliant
This COBIT indicator is indirectly covered by the following two AGIT indicators:

- AG1-4: Error Density;
- AG1-5: Costs of Rework.

AGIT uses the number of errors reported by the user in a fixed period after release as well as classification of tasks in the Sprint Backlog according to the type of work performed (development, testing, rework due to the change in requirements, rework due to error reported by the customer, etc.).

AI6-1: Number of disruptions or data errors caused by inaccurate specifications or incomplete impact assessment
Status: Compliant
This COBIT indicator is indirectly covered by the following two AGIT indicators:

- AG1-4: Error Density;
- AG1-5: Costs of Rework.

In this case the rework is due to inaccurate specifications or incomplete impact assessment.

AI6-2: Amount of application or infrastructure rework caused by inadequate change specifications
Status: Compliant
This COBIT indicator is indirectly covered by the following two AGIT indicators:

- AG1-4: Error Density;
- AG1-5: Costs of Rework.

In this case the rework is due to inadequate change specifications.

AI6-3: Percent of changes that follow formal change control processes
Status: Compliant
This COBIT indicator is covered by the Daily Scrum meeting, when changes are managed according to Scrum.

AI7-1: Amount of application downtime or number of data fixes caused by inadequate testing
Status: Compliant
This COBIT indicator is indirectly covered by the following two AGIT indicators:

- AG1-4: Error Density;
- AG1-5: Costs of Rework.

In this case the rework is due to inadequate testing.

AI7-2: Percent of systems that meet expected benefits as measured by the post-implementation process
Status: Compliant
This COBIT indicator can be mapped to AGIT indicator AG4-1: "Qualitative Evaluation of Customer Satisfaction using Criteria".

AI7-3: Percent of projects with a documented and approved testing plan
Status: Compliant
This COBIT indicator is covered by the Sprint Backlog, under the condition that the tasks in the Sprint Backlog include testing of components. This way AI7-3 can be covered by AGIT without violating the principles of agility.

Table 4: COBIT Indicators for System Development Life Cycle – Domain AI

COBIT Indicator	*Status*	AGIT Indicator/Scrum requirement
AI1-1: Number of projects where stated benefits were not achieved due to incorrect feasibility assumptions	***Partly compliant***	**AG1-6:** Fulfilment of scope
AI-2: Percent of feasibility studies signed off on by the business process owner	*Compliant*	*Scrum: Sprint planning meeting*
AI1-3, AI2-2: Percent of users satisfied with functionality delivered	*Compliant*	**AG4-1:** Qualitative evaluation of customer satisfaction
AI2-1: Number of production problems per application causing visible downtime	*Compliant*	**AG1-4:** Error density **AG1-5:** Costs of rework
AI6-1: Number of disruptions or data errors caused by inaccurate specifications or incomplete impact assessment	*Compliant*	**AG1-4:** Error density **AG1-5:** Costs of rework

AI6-2: Amount of application or infrastructure rework caused by inadequate change specifications	*Compliant*	**AG1-4:** Error density **AG1-5:** Costs of rework
AI6-3: Percent of changes that follow formal change control processes	*Compliant*	*Scrum: Daily Scrum meeting*
AI7-1: Amount of application downtime or number of data fixes caused by inadequate testing	*Compliant*	**AG1-4:** Error density **AG1-5:** Costs of rework
AI7-2: Percent of systems that meet expected benefits as measured by the post-implementation process	*Compliant*	**AG4-1:** Qualitative evaluation of customer satisfaction
AI7-3: Percent of projects with a documented and approved testing plan	*Compliant*	*Scrum: Sprint Backlog*

5.2.3 Domain Deliver and Support (DS)

The results for domain DS are shown in Table 5.

DS5-1: Number of incidents damaging the organisation's reputation with the public, **DS5-2**: Number of systems where security requirements are not met, **DS5-3:** Number of violations in segregation of duties

Status: Compliant

AGIT indicator AG1-4: "Qualitative Evaluation of Customer Satisfaction using Criteria" is compliant with these COBIT indicators; under the condition that questionnaire includes questions regarding security requirements that are result of development activities.

DS10-1: Number of recurring problems with an impact on the business

Status: Compliant

AGIT indicator AG1-4: "Qualitative Evaluation of Customer Satisfaction using Criteria" is compliant with this COBIT indicator.

DS10-2: Percent of problems resolved within the required time period

Status: Compliant

This COBIT indicator is covered by the following two AGIT indicators:

- AG3-1: Average Number of Impediments per Task/Sprint/Team;
- AG3-2: Mean Time for Resolving an Impediment (at Task/Sprint/Team level).

DS10-3: Frequency of reports or updates to an ongoing problem, based on the problem severity

Status: Compliant

This COBIT indicator is covered by the ScrumMaster role, so the frequency is daily.

Table 5: COBIT Indicators for System Development Life Cycle – Domain DS

COBIT Indicator	***Status***	**AGIT Indicator/Scrum requirement**
DS5-1: Number of incidents damaging the organisation's reputation in public	*Compliant*	**AG4-1:** Qualitative evaluation of customer satisfaction
DS5-2: Number of systems where security requirements are not met	*Compliant*	**AG4-1:** Qualitative evaluation of customer satisfaction
DS5-3: Number of violations in segregation of duties	*Compliant*	**AG4-1:** Qualitative evaluation of customer satisfaction
DS10-1: Number of recurring problems with an impact on the business	*Compliant*	**AG4-1:** Qualitative evaluation of customer satisfaction
DS10-2: Percent of problems resolved within the required time period	*Compliant*	**AG3-1:** Average number of impediments **AG3-2:** Mean time for resolving an impediment
DS10-3: Frequency of reports or updates to an ongoing problem, based on the problem severity	*Compliant*	*Scrum: Daily Scrum meeting*

5.3 Interpretation of the Comparison Results

The results, presented in Tables 2-5 show, that the majority or 22 of 26 (85%) applicable COBIT indicators are covered by the AGIT indicators (PO7-1, PO8-1, PO8-2, PO8-3, PO10-1, PO10-3, AI1-2, AI1-3, AI2-1, AI2-2, AI6-1, AI6-2, AI6-3, AI7-1, AI7-2, AI7-3, DS5-1, DS5-2, DS5-3, DS10-1, DS10-2, DS10-3).

Only 3 of 26 (11%) applicable COBIT indicators can be partly mapped to AGIT indicators (PO7-2, PO10-2, AI1-1).

Finally, only 1 of 26 (4%) applicable COBIT indicators is not included in AGIT model (PO7-3).

Since 85% of applicable COBIT indicators are compliant with AGIT indicators, we can say that the AGIT model almost completely satisfies COBIT criteria.

Non-compliant COBIT indicator does not depend on the software development method (in our case Scrum), but is related to the human resources strategy (PO7-3: Percent of IT personnel certified to job needs).

The same applies to the partly compliant indicators, that relate to PO7-2: IT personnel turnover (partly covered by AG2-1: The average amount of overtime, AG2-2: The average number of projects the employees work in parallel and AG2-3: Qualitative evaluation of working conditions), PO10-2: Percent of projects receiving post-implementation reviews (partly covered by Sprint retrospective meeting) and AI1-1: Number of projects where stated benefits were not achieved due to incorrect feasibility assumptions (partly covered by AG1-6: Fulfilment of scope).

6 Further AGIT Development

6.1 Adjusting AGIT Indicators

In this section we introduce our solution for non-compliant and partly compliant indicators.

Non-compliant COBIT indicator PO7-3: "Percent of IT personnel certified to job needs" can be calculated by keeping records of the team members' certificates. We have decided to add this indicator to AGIT model.

Partly compliant COBIT indicator PO7-2: "IT personnel turnover" can be calculated by keeping records not only on the number of all developers, as recommended in AGIT model, but also on the number of the developers that left the company and the number of the new developers (in the certain period of time). Partly compliant COBIT indicator PO10-2: "Percent of projects receiving post-implementation reviews" (partly covered by Sprint retrospective meeting) can be calculated by keeping records abut post-implementation reviews. The last partly compliant COBIT indicator AI1-1: "Number of projects where stated benefits were not achieved due to incorrect feasibility assumptions" can be calculated by introducing classification of the causes of the non-completion of the tasks for the indicator "fulfilment of scope" (one of the possible causes could be incorrect feasibility assumptions). We have decided to add these three indicators to AGIT model.

The final result was the addition of the four new COBIT indicators to AGIT model.

For the stakeholder "Team Members" we have added the following two indicators:

- **PO7-2**: "IT personnel turnover" and
- **PO7-3**:"Percent of IT personnel certified to job needs".

For the stakeholder "IT Management" we have added the following two indicators:

- **PO10-2**: "Percent of projects receiving post-implementation reviews" and
- **AI1-1**: "Number of projects where stated benefits were not achieved due to incorrect feasibility assumptions".

6.2 IT Balanced Scorecard

After adjusting the indicators of the AGIT model, we wanted to adjust the presentation of its structure as well. In order to increase its clarity and familiarity to the executive management, we have decided to use the balanced scorecard form of presentation [11].

The primary goal of BSC is to transform strategy into action and allow management to monitor the implementation of the strategy. BSC includes performance measurement from four perspectives [9], as shown in Fig. 3.

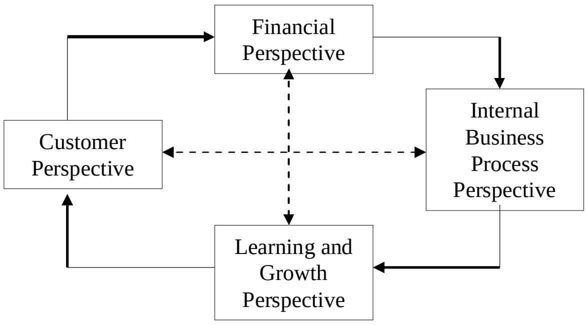

Financial Perspective
Internal Business Process Perspective
Customer Perspective
Learning and Growth Perspective

Fig. 3: BSC (adjusted from [11])

IT has its own balanced scorecard (IT BSC), with redefined four perspectives ([8],[26]):

- Enterprise contribution: How do business executives view the IT department?
- User orientation: How do users view the IT department?
- Operational excellence: How effective and efficient are the IT processes?
- Future orientation: How well is IT positioned to meet future needs?

BSC and IT BSC (Fig. 4) enables reader to capture all four perspectives at once. We find that this presentation is clear and simple, and we prefer this approach to the sequential table approach used so far. Therefore we have decided to harmonize the presentation of adjusted AGIT model structure with the IT BSC presentation form.

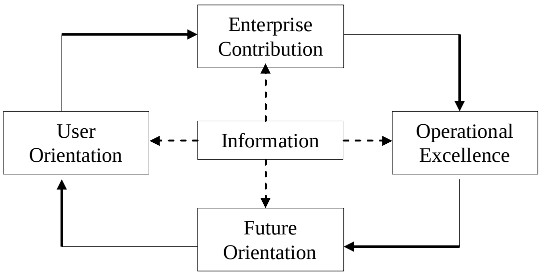


Fig. 4: IT BSC (adjusted from [8])

6.3 Solution for AGIT-0.3

For the purpose of this paper we have given the adjusted AGIT model name AGIT-0.3, assuming that the previous two versions are:

- AGIT-0.1: the first published version of the model in which the key indicators were presented [15];
- AGIT-0.2: the second published version of the model that included presentation of the data repository design [16].

One of our criteria for good measurement system is simplicity and flexibility. In order to increase simplicity of our model we have mapped our four stakeholders to the four perspectives of IT BSC:

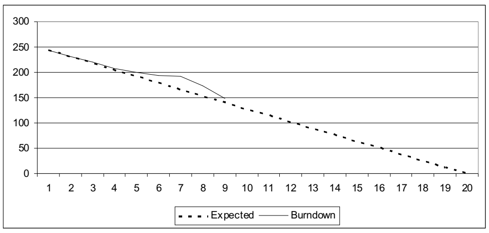
- IT Management - Enterprise contribution;
- Customer - User orientation;
- ScrumMaster - Operational excellence;
- TeamMembers - Future orientation.

The adjusted indicators and adjusted presentation can be seen in the Table 6. This way, each stakeholder can monitor the implementation of its goals in easy and simple way, in compliance with Agility Principles.

Table 6: AGIT-0.3 Indicators for Scrum-based Software Development Process

AG1: IT Management	
Timely Information on Project	**AG1-1:** Work Effectiveness
	AG1-2: Schedule Performance Index (SPI)
	AG1-3: Cost Performance Index of Labor Costs (CPI)
Quality Improvement	**AG1-4:** Error Density
	AG1-5: Costs of Rework
	AG1-6: Fulfilment of Scope
	PO10-2: *Percent of projects receiving post-implementation review*
	AI1-1: *Number of projects where stated benefits were not achieved due to incorrect feasibility assumptions*

AG4: Customers	
Customer Satisfaction	**AG4-1:** Qualitative Evaluation of Customer Satisfaction
	AG1-4: Error Density
	AG1-6: Fulfilment of Scope



AG3: ScrumMaster	
Efficient Impediments Resolution	**AG3-1:** Average Number of Impediments per Task/Sprint/Team
	AG3-2: Mean Time for Resolving an Impediment (at Task/Sprint/Team level)

AG2: Team Members	
Job Satisfaction	**AG2-1:** The average amount of overtime
	AG2-2: The average number of projects the Employees work in parallel
	AG2-3: Qualitative evaluation of working conditions
	PO7-2: *IT personnel turnover*
	PO7-3: *Percent of IT personnel certified according to job needs*

7 Conclusion

Measurement of agile software development has been explored by many authors ([1], [27], [4]). Our previous research in this area is summarized in the AGIT (AGIle software developmenT) model, which includes basic indicators for measurement of Scrum-based software development [15], CMMI Measurement and Analysis Practices for Scrum-based Software Development Process [17] and a description of corresponding measurement repository [16].

The aim of this paper was to assess AGIT model by determining the level of compliance with the information systems auditing criteria, as described in COBIT (Control Objectives for Information and Related Technology) [8], the IT governance framework commonly used for IT governance implementation and assessment.

After short introduction of the Scrum, AGIT and COBIT models, we compared the appropriate indicators from COBIT model for system development life cycle with the indicators proposed in AGIT model. For this purpose we used the selection of COBIT processes as defined in the information systems auditing guideline G23 [6], slightly extended by selection for agile development by [5] and additional two processes that, in our opinion, refer to agility. We compared 26 indicators for the selected COBIT processes with 12 indicators from AGIT model. The results of comparison were one non-compliance and three partial-compliances, which did not depend on the software development method. Therefore we concluded that model AGIT almost completely satisfies COBIT criteria. Then we introduced our solution for these non/partly compliances in the new version of the model, named AGIT-0.3. In the end we simplified the presentation of AGIT-0.3 model structure using BSC cross form.

We hope that the results of this paper contribute to the quality and clarity of AGIT model. In the near future we plan to map this model to other comparable and accepted models.

References:

[1] B. Barton, K. Schwaber, D. Rawsthorne, Reporting Scrum Project Progress to Executive Management through Metrics (2005, January) *http://www.scrumalliance.org/system/resource/file/21/Reporting_Scrum_Project_Progress_to_Executive_Management_through_Metrics.pdf* >

[2] B. Barton, E. Campbell, Implementing a Professional Services Organization Using Type C Scrum, *Proceedings of 40th Annual Hawaii International Conference on System Sciences (HICSS07)*, pp. 1-9 *<http://csdl2.computer.org/comp/proceedings/hicss/2008/3075/00/30750466.pdf>*

[3] M. Ceschi et al., Project Management in Plan-Based and Agile Companies, *IEEE Software,* May/June 2005, pp. 21-27.

[4] E. Damiani, A. Colombo, F. Frati, C. Bellettini, A Metamodel for Modeling and Measuring Scrum Development Process, *Lecture Notes in Computer Science 4536,* Springer-Verlag Berlin Heidelberg, 2007, pp. 74-83.

[5] S. Gupta: SOX Compliant Agile Processes, *Proceedings of AGILE 2008 Conference (AGILE'08)*, pp. 140-143.

[6] ISACA, *IS Standards, Guidelines and Procedures for Auditing and Control Professionals,* Information Systems Audit and Control Association, Rolling Meadows, 2008.

[7] ITAF, *A Professional Practices Framework for IT Assurance*, Information Systems Audit and Control Association, Rolling Meadows, 2008.

[8] ITGI, *COBIT v4.1*, Information Technology Governance Institute, Rolling Meadows, 2007.

[9] ITGI, *IT Governance Implementation Guide: Using COBIT and ValIT, 2nd Edition*, Information Technology Governance Institute, Rolling Meadows, 2007.

[10] C. R. Jakobsen, K. A. Johnson, Mature Agile with a twist of CMMI, *Proceedings of AGILE 2008 Conference (AGILE'08)*, pp. 212-217.

[11] R. S. Kaplan, D. P. Norton, *The Balanced Scorecard: translating strategy into action*, President and Fellows of Harvard College, 1996.

[12] P. Kueng, Process performance measurement system: a tool to support process-based organizations, *Total Quality Management*, Vol. 11, No. 1, 2000, pp. 67-85.

[13] V. Mahnic, S. Drnovscek, Agile Software Project Management with Scrum, *EUNIS 2005 Conference – Session papers and tutorial abstracts*, University of Manchester, June 2005, CD-ROM. *<http://www.mc.manchester.ac.uk/eunis2005/medialibrary/papers/paper_194.pdf>*

[14] V. Mahnic, S. Drnovscek, Introducing agile methods in the development of university information systems. *Proceedings of the 12th International Conference of European University Information Systems EUNIS 2006,* Tartu, June 2006 (2006), pp. 61-68.

[15] V. Mahnic, I. Vrana, Using stakeholder driven process performance measurement for monitoring the performance of a Scrum based software development process, *Electrotechnical Review*, Ljubljana, Vol. 74, No. 5, 2007, pp. 241-247.

[16] V. Mahnic, N. Zabkar, Measurement repository for Scrum-based software development process, *Modern topics of computer science : proceedings of the 2nd WSEAS International Conference on Computer Engineering and Applications (CEA'08)*, Acapulco, Mexico, January 25-27, 2008 pp. 23-28.

[17] V. Mahnic, N. Zabkar, Introducing CMMI Measurement and Analysis Practices into Scrum-based Software Development Process, *International Journal of Mathematics and Computers in Simulation*, Issue 1, Volume 1, 2007, pp. 65-72.

[18] Manifesto for Agile Software Development, 2001 <*http://www.agilemanifesto.org/*>

[19] C. Mann, F. Maurer, A Case Study on the Impact of Scrum on Overtime and Customer Satisfaction, *Proceedings of the Agile Development Conference (ADC'05)*, pp. 70-79.

[20] E. Markopoulos, J. Bilbao, E. Bravo, T. Stoilov, T. E. J. Vos, C. Figa´Talamanca, K. Reschwamm, Project Management Stage Mutations within Agile Methodological Framework Process Transformations, *WSEAS TRANSACTIONS on INFORMATION SCIENCE & APPLICATIONS*, Issue 5, Volume 5, 2008, pp. 776-785.

[21] PMI, A Guide to Project Management Body of Knowledge, third edition (PMBOK Guide), Project Management Institute, Inc., Four Campus Boulevard, Newtown Square, Pensylvania, 2004.

[22] B. Schatz, I. Abdelshafi, Primavera Gets Agile: A Successful Transition to Agile Development, *IEEE Software*, May/June 2005, pp. 36-42.

[23] K. Schwaber, Agile Project Management with Scrum, *Microsoft Press*, 2004.

[24] C. Schwaber et al., The Truth About Agile Processes (2007, December 3).
<*www.forrester.com*>

[25] M. Spremic, M. Popovic, Emerging issues in IT Governance: implementing the corporate IT risks management model, *WSEAS Transactions on Systems,* Issue 3, Volume 7, March 2008, pp. 219-228.

[26] M. Spremic, Z. Zmirak, K. Kraljevic, Evolving IT Governance Model – Research Study on Croatian Large Companies, *WSEAS Transactions on Business and Economics*, Issue 5, Volume 5, May 2008, pp. 244-253.

[27] T. Sulaiman, B. Barton, T. Blackburn, AgileEVM - Earned Value Management in Scrum Projects, *Proceedings of AGILE 2006 Conference (AGILE'06)*, pp. 7-16.

[28] J. Sutherland et al., Scrum and CMMI Level 5: The Magic Potion for Code Warriors, *Proceedings of AGILE 2007*, pp. 272-278

[29] J. Sutherland, K. Schwaber, The Scrum Papers: Nuts, Bolts, and Origins of an Agile Process,
Draft 10/14/2007
<*www.jeffsutherland.com/scrum/ScrumPapers.pdf*>

[30] B. Upender, Staying Agile in Government Software Projects, *Proceedings of the Agile Development Conference (ADC'05)*, pp. 153-159.

[31] T. Wailgum, From Here to Agility (2007, December 3).
< *http://www.cio.com*>

[32] N. Zabkar, V. Mahnic, Using COBIT Model in Software Development, *Proceedings of Slovenian Informatics Conference 2005 (DSI 2005)*, April 13-15, 2005, Portoroz, Slovenia, pp. 257-262