International Journal of Information Technologies and Systems Approach
Volume 14 • Issue 1 • January-June 2021

Optimization of Cyber Defense Exercises Using Balanced Software Development Methodology

Radek Ošlejšek, Masaryk University, Czech Republic
Tomáš Pitner, Masaryk University, Czech Republic

ABSTRACT

Cyber defense exercises (CDXs) represent an effective way to train cybersecurity experts. However, their development is lengthy and expensive. The reason lies in current practice where the CDX life cycle is not sufficiently mapped and formalized, and then exercises are developed ad-hoc. However, the CDX development shares many aspects with software development, especially with ERP systems. This paper presents a generic CDX development method that has been derived from existing CDX life cycles using the SPEM standard meta-model. The analysis of the method revealed bottlenecks in the CDX development process. Observations made from the analysis and discussed in the paper indicate that the organization of CDXs can be significantly optimized by applying a balanced mixed approach with agile preparation and plan-driven disciplined evaluation.

KEYWORDS

Agile, CDX, Cyber Range, Cybersecurity, Development Method, Discipline, Disciplined, Learning, Phase, SPEM, Training

INTRODUCTION

A shortage of cybersecurity workforce poses a critical danger for current companies and nations (Restuccia, 2015; Ministr & Pitner, 2019). As modern society is exposed to the increasing number of cyber threats, there is a growing need to train new cybersecurity experts.

Cyber defense exercises (CDX) (Díez, Pereira, Merino, Suárez, & Juan, 2015) represent a popular type of training that aims to fill this skill gap. They have been traditionally organized by military and governmental agencies (Petullo, Moses, Klimkowski, Hand, & Olson, 2016). CDXs emphasize realistic training scenarios that authentically mimic the operational environment of a real organization (Eagle, 2013). For these reasons, every new CDX is unique. Its preparation requires a considerable amount of skills and workforce. It takes months to prepare and organize a new CDX event with a substantial number of people being involved. These circumstances make the realization of effective hands-on training programs extremely demanding, costly, and with a high risk of failure. One of the reasons is the lack of development methodology when the development of CDXs is rather ad-hoc and loosely driven nowadays.

CDXs are in many aspects similar to traditional software projects. Especially, it is possible to find the parallel between their life cycle and the life cycle of ERP systems (Botta-Genoulaz, Millet, & Grabot, 2005) – systems which are composed of existing modules that have to be adapted to customer's business processes, deployed at customer's site, and maintained. In the cybersecurity

DOI: 10.4018/IJITSA.2021010108

domain, ERP systems are replaced with so-called *cyber ranges*. They represent complex software and hardware environments providing isolated computer networks where cybersecurity exercises can be safely organized without the danger of threatening real users or IT infrastructure. Similarly to the ERP systems that have to be adjusted for individual customers, also cyber ranges have to be adapted, instantiated and configured for each CDX. However, business domains differ. While ERP systems track business resources (e.g., cash, material, or production capacity) to support planning, purchasing, and sale, cyber ranges are designed to track vulnerabilities, attacks, network services, and other cybersecurity aspects to support learning processes.

As the parallel between cyber ranges and ERP systems is evident, the utilization of the software development methods for CDX preparation seems to be meaningful. Software companies struggle to optimize the provision of IT services by forcing developers to seek better methods for their business informatics management (Voříšek, Pour, & Buchalcevová, 2015). In the same way, this paper aims to improve effectiveness and reduce the cost of CDX development by searching for iterative and incremental approaches (Larman & Basili, 2003; Bar-Yam, 2003) that could help to deal with the complexity and rapid changes emerging in CDX development and management.

This paper can be seen as a Design Science Research (DSR) with the exaptation type of contribution (Gregor & Hevner, 2013). Exaptation research extends known solutions to new problems. It is characterized by low maturity of the application domain and high solution maturity. The application domain of this research is the CDX development. As a solution for its low efficiency, the authors aim to use agile or disciplined principles.

This paper contributes to two types of DSR knowledge: prescriptive and descriptive (Mokyr et al., 2002). As a contribution to prescriptive knowledge, a CDX development method is proposed. It is built on the application of the *Software & Systems Process Engineering Metamodel – SPEM* (Object Management Group, Inc, 2008) methodology on existing CDX life cycles. This method then serves as a conceptual framework for further analysis. As a contribution to descriptive knowledge, key bottlenecks of the CDX development method are identified. Their reduction using either agile or disciplined principles is discussed.

RELATED WORK

The use of agile methodologies has increased significantly over the past decades (Dingsøyr, Nerur, Balijepally, & Moe, 2012; Hoda, Salleh, & Grundy, 2018), promoting the value of the human-centric software development process. However, agile development suffers from many limitations (Misra, Kumar, Kumar, Fantazy, & Akhter, 2012; Turk, France, & Rumpe, 2014), and then it is not suitable for all types of projects (Ghayyur et al., 2018).

On the contrary, traditional plan-driven methods (also call disciplined methods) like Rational Unified Process (Kruchten, 2004) comes from the assumption that planning and documentation is the key to successful project management and development. They focus on repeatability, predictability, verification, and validation. However, these features can make plan-driven methods too rigid and hardly adapting to changing requirements.

Balanced (also called hybrid) methods represent a mixture of both the worlds. In (Boehm & Turner, 2003), the authors provide a comprehensive survey on agile and disciplined methods and discuss the ways of their balancing. They conclude that *"there is no agile or plan-driven method silver bullet"*. Hybrid models combining agile and traditional development can be found in (Zaki & Moawad, 2010; Galván-Cruz, Mora, & O'Connor, 2017). In (Imani, Nakano, & Anantatmula, 2017), the authors show that the hybrid approach should be more scalable than the agile methods and that the hybrid approach can provide better cost-benefit ratios compared to the traditional plan-driven methods.

As the development process of CDXs is ad-hoc and informally driven in current practice, the character of the CDX life cycle is unknown. The authors did not find any work dealing with the application of either agile, disciplined, or balanced methodology on CDX development. Therefore,

current knowledge in the field of project management and system development is used to define a CDX development method and discuss its agile vs. disciplined characteristics. This paper builds on the study of existing cyber defense exercises. The literature survey revealed three key papers dealing with organizational aspects of CDXs.

The Cyber Exercise Playbook (Kick, 2014) defines three phases of CDX development and describes user roles participating in the life cycle. The playbook focuses primarily on the planning phase, which is organized as a series of five consecutive meetings. This model is also discussed and summarized in (Seker & Ozbenli, 2018).

The CyberRX Playbook (Alliance, 2015) introduces four phases. This work emphasizes the need for regular improvement of the cybersecurity program via internal lessons learned. Putting the stress on repeatability and continual improvement puts additional demands on the life cycle.

Probably the most detailed life cycle is discussed in (Vykopal, Vizváry, Ošlejšek, Čeleda, & Tovarňák, 2017). Their model is based on the experience from the organization of the Cyber Czech exercise. This paper describes the responsibilities of user roles in five phases and also describes significant outputs. Bottlenecks of the development process are discussed, as well. The time and workforce required for the development are identified as critical problems.

Inefficiencies in the CDX life cycle are addressed also in (Yamin & Katt, 2018). According to the authors, *"cyber-security exercises are a good tool for cyber-security skill development, but the inefficiencies in cyber-security exercise development and execution life cycle limit its ability to be widely used for cyber-security skill development"*.

Although the core of different life cycles is similar, they vary in many details like the number and names of phases, or names of roles and their responsibilities. They also differ in the level of detail in which the discussion is held. Therefore, it is difficult to generalize them, derive unified concepts, and identify bottlenecks in the workflow that could be eliminated. This paper struggled to fill this gap by providing a unified development method.

Apart from studying CDX life cycles, the researchers also focused on the analysis of existing cyber ranges, as their features can significantly affect CDX development. The development of cyber ranges has seen a large increase in recent years. There is an extensive survey of state-of-the-art cyber ranges and testbeds in (Davis & Magrath, 2013).

Although there are many cyber ranges available worldwide, e.g., Michigan Cyber Range (MCR, n.d.), SimSpace Cyber Range (Rossey, n.d.), or EDURange (EDURange, n.d.), there are not many sources publicly available providing sufficient details about their features and architecture. It is because the cybersecurity domain represents a sensitive area sharing many similarities with military or intelligence services, in which many sources are secret or restricted.

Fortunately, exceptions exist. One very popular cyber range is DETER/DeterLab (Mirkovic et al., 2010; Benzel, 2011), which was started to advance cybersecurity research and education. The description of the architecture, features, and operation can be found also for CyRIS (Pham, Tang, Chinen, & Beuran, 2016), CyTrONE (Beuran et al., 2018), NCR (Ferguson, Tall, & Olsen, 2014), and KYPO (Vykopal, Ošlejšek, Čeleda, Vizváry, & Tovarňák, 2017).

Exploration of these cyber ranges shows that, despite some differences, they share many common features and concepts. This paper primarily builds on the KYPO cyber range platform (Vykopal, Ošlejšek, Čeleda, Vizváry, & Tovarňák, 2017), in whose development are the authors directly involved. However, the presented observations and features related to CDX development can be generalized and valid for all similar modern platforms.

RESEARCH METHOD

All the conceptual papers that have been found during the literature survey (Kick, 2014; Alliance, 2015; Díez et al., 2015; Vykopal, Vizváry, Ošlejšek, Čeleda, & Tovarňák, 2017) divide the CDX

life cycle into several consecutive phases ending with milestones. This fact suggests that the global character of CDXs is rather disciplined.

Based on this observation, SPEM (Object Management Group, Inc, 2008) was chosen as a meta-model to be used to analyze the CDX life cycle in detail and provide a methodological view of its development. SPEM can be considered as a continuous evolution of the IBM RUP meta-model (Shuja & Krebs, 2007), where the division of the development into consecutive phases play an important role. This process-oriented meta-model is often used as a baseline framework for the conceptualization of software engineering processes (Ruiz-Rube, Dodero, Palomo-Duarte, Ruiz, & Gawn, 2013; Baumgarten, Rosinger, Todino, & de Juan Marín, 2015; Diaw, Cisse, & Bah, 2017).

SPEM provides conceptualization from different perspectives. In this paper, the SPEM is used to comprehend the rationale of the CDX development process and to create a model suitable for the analysis of bottlenecks. This work utilizes the following selected elements of the SPEM 2.0 Base Plug-in (Object Management Group, Inc, 2008, p. 155) to get a model with a convenient level of abstraction:

- **Activity kinds:** CDX-specific phases and milestones from CDX life cycles were defined. Basic activities were derived, problems in their implementation were identified. Possible decomposition of phases into iterations was analyzed.
- **Work product kinds:** The main problem of the CDX organization and preparation is low efficiency. Overtaking this bottleneck requires improving the repeatability of the CDX development processes and struggling for its maximal automation. Therefore, during the modeling, attention was payed primarily on artifacts that represent tangible elements like documents or formalized knowledge bases.
- **Work product relationship kinds:** Decomposition was omitted to keep the modeling and analysis on a suitable level of abstraction. Instead, this paper deals with a flat model of dependencies between artifacts (also referred to as impacted by relationships in the SPEM meta-model).
- **Category kinds:** Roles were derived from the skills, competencies, and responsibilities of individuals identified in the CDX life cycle. Disciplines of SW development, especially of the ERP systems, were identified and adapted to the specifics of the CDX life cycle. Then, the activities were further elaborated considering roles, work products, and work product relationships involved in them.

The process of the SPEM application was iterative. Section *CDX Development Method* of this paper corresponds to the *process structure* perspective of the SPEM (Object Management Group, Inc, 2008, p. 43), where roles and phases are discussed. The *process with methods* perspective (Object Management Group, Inc, 2008, p. 95), i.e., the disciplines, artifacts, and their dependencies, is discussed in the subsequent section *Detailed Discussion of Disciplines*. The authors drew from the existing models of CDX life cycles, but also from the 6-years old experience of the authors with the development of the KYPO cyber range and organization the Cyber Czech CDX. The model was discussed with domain experts – organizers of the Cyber Czech CDX.

The high-level model resulting from the application of the SPEM meta-model on CDX life cycles is shown in Figure 1. The scheme uses two dimensions to capture the approximate effort needed by development activities. The time dimension (the x-axis) splits the CDX life cycle into phases, while the workflow dimension (the y-axis) includes working activities called disciplines.

Once the conceptual model was available, the research continued in the analysis of the application of agile or disciplined approaches to the critical parts of the CDX development method. This process consisted of two stages. First, problems and possible solutions to the four development phases were identified regardless of discipline. They are described as part of section *CDX Development Method*. Then, the analysis of activities within individual disciplines and critical phases was conducted.

Obtained dependency models were used to formulate recommendations for using agile or disciplined approaches. These per-discipline observations are described within section *Detailed Discussion of Disciplines*.

CDX DEVELOPMENT METHOD

All the conceptual papers that have been found during the literature survey (Kick, 2014; Alliance, 2015; Díez et al., 2015; Vykopal, Vizváry, Ošlejšek, Čeleda, & Tovarňák, 2017) divide the CDX life cycle into several consecutive.

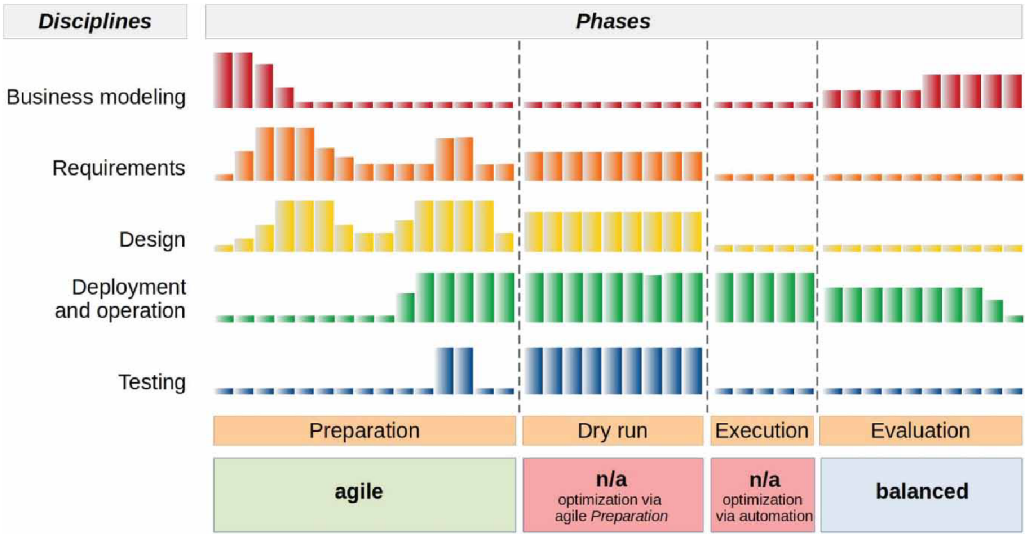
This section focuses primarily on the time dimension of the CDX development method (the x-axis in Figure 1). The goal is to describe basic characteristics of phases regardless of disciplines. First, roles involved in phases are introduced.

Roles

Methods of traditional software-system development introduce standard roles for people involved in the process, like analysts, developers, testers, or stake-holders. However, the development of a CDX is specific. It is more similar to adjusting an existing generic system for a particular customer or business domain rather than developing a bright new system from scratch. Archetypal roles defined in this section come from steady terminology established in the field of cybersecurity education where color teams are used to differentiate the responsibilities of people in the exercise (Kick, 2014; Vykopal, Vizváry, Ošlejšek, Čeleda, & Tovarňák, 2017; Brilingait˙e, Bukauskas, & Juozapavičius, 2019). Apart from these "color-named" teams, additional roles are introduced so that the entire CDX life cycle is covered.

- **Stakeholder:** A representative of an organization whose needs are to be met by the exercise. Cyber defense exercises represent big expensive events that are usually organized on the request of specific customers willing to train their experts. Often, these customers represent bigger commercial subjects operating critical infrastructures, e.g., energy distributors, governmental authorities, ministries, or national security agencies. Stakeholders are always involved in the CDX life cycle. Although the level of their involvement may differ, they often intensively participate

Figure 1. Development method for cyber defense exercises: It consists of four phases and five disciplines. Bar charts suggest approximate work effort required to organize a cyber defense exercise. Observed characteristics of the phases are provided at the bottom.



in all stages of the CDX life cycle. Stakeholders are usually involved in the content preparation, they are presented as observers during the training event, and they want to be informed about the learning impact on the trainees. On the other hand, some stakeholders perceive CDX as a service and rely on the CDX organization teams that they do the best.

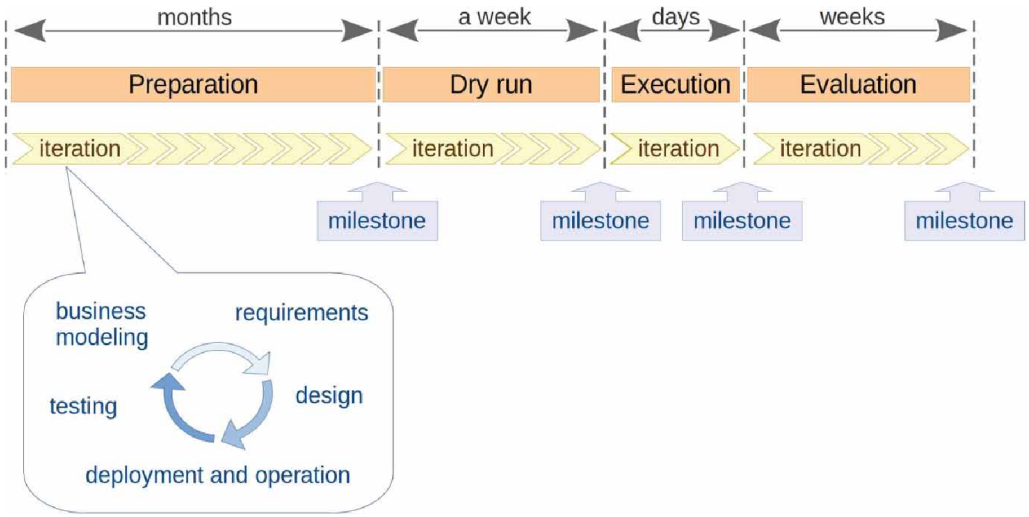
- **Training Expert:** An expert skilled in training people. The ultimate goal of any CDX is to train participants properly. However, the impact of the training on participants can be affected by many factors. Training experts are experienced in organizing cybersecurity training sessions. They are able to consider the learning aspects of the exercise. They act as mediators and coordinators between stakeholders and IT experts (members of red, white, and green teams – see below) aiming to reflect their ideas and expectations in the exercise.
- **Blue Team:** A group of trainees that cooperate during the exercise to defend a computer network against attackers. Blue teams are usually composed of cybersecurity practitioners like network administrators whose motivation is to train and enhance their skills via CDX. Their goal is to secure an entrusted computer network and defend it against attacks of the red team during the CDX training session. A typical CDX event is organized for several (4-5) blue teams, each of which consists of a few (4-5) participants. All blue teams manage identical network infrastructure and face the same attacks of the red team. Members of blue teams do not participate in the development process of CDX. Instead, they can be seen as end-users of the final product.
- **Red Team:** A group of people technically skilled and authorized to conduct cybersecurity attacks. During the CDX development, they are responsible for the definition of meaningful attack plans, vulnerabilities, and attack vectors. During the CDX training session, they follow the attack plan to exploit vulnerabilities left in computer networks of blue teams. Based on the success of attacks, the red team assigns penalties to blue teams.
- **Green Team:** A group of responsible for the cyber range management. Hands-on training sessions are organized in complex underlying infrastructures that have many technical limitations. Knowledge of these technical aspects is necessary during the CDX development to moderate expectations of stakeholders with regard to possibilities of the cyber range. The green team also configures the cyber range for particular CDX. Moreover, members of the team play an important role during the training sessions. They monitor the infrastructure, fixes occasional crashes and infrastructure issues, and revert networks of blue teams to a functional state if they unintentionally cut off the access to the network on their firewall, for instance.
- **White Team:** The goal of CDXs is to train soft skills in addition to technical expertise. A white team, therefore, simulates media requesting reports from blue teams, regular users of defended networks, law enforcement agencies, and other fictitious users that the blue teams have to interact with. Moreover, they act as judges, enforce the rules of the exercise, observe the exercise, score blue teams, and ensure that the competition runs fairly. During the CDX development, they are responsible for the definition of non-technical content of the exercise, like a background story, or tasks of fictitious users with corresponding penalties.

Phases

According to the SPEM standard, a phase represents a significant period in a project, ending with a major management checkpoint, milestone, or set of deliverables. Phases are activities that are not expected to be repeatable during the project life cycle. Every phase can be divided into multiple iterations, as depicted in Figure 2.

The CDX development consists of four phases that have been derived from existing CDX life cycles. Their description stays at the conceptual level without going into the details of outputs and activities. These details are described later as part of the discussion of specific disciplines. The text follows the terminology introduced by (Vykopal, Vizváry, Ošlejšek, Čeleda, & Tovarňák, 2017) and (Kick, 2014).

Figure 2. CDX phases and their relation to iterations and milestones



For each phase, a brief description is provided and accompanied by an expected milestone. The milestone captures key achievements that are to be accomplished at the end of each phase. Also, troubles and difficulties related to each phase are summarized. The application of the disciplined or agile approach is discussed as well.

Phase 1: Preparation

Preparation is the first phase of any new CDX. The goal is to define the content of the exercise, specify technical requirements, allocate resources, test coherence of the training scenario, and verify the functionality of the cyber range infrastructure.

- **Milestone:** The cyber range is completely instantiated, configured, and ready for use. Scenarios prescribing the expected steps and tasks of *red* and *white teams* are completed, their coherence and meaningfulness are verified.
- **Identified troubles:** The long-term experience of the authors with developing and organizing CDXs shows that the preparation phase is extremely demanding. It takes several months to prepare a new CDX either from scratch or by significantly changing an existing scenario. Moreover, a lot of people have to be involved in this process and coordinated. These aspects lead to the high rate of errors and logical inconsistencies that have to be revealed and repaired in the later *dry run* phase. These aspects make the preparation phase very expensive.
- **Solution:** CDX preparation is very creative process with unclear requirements at the beginning that have to be clarified by intensive discussion and cooperation of many specialists (*training experts, red team, white team, green team*). Tight cooperation with *stakeholders* and partially with prospective trainees (i.e., *blue teams* participating in the prerequisite testing) is also necessary. Moreover, the budget for the exercise and the schedule of its preparation are usually appointed in advance. These features dominate in agile methods, and then the application of an agile approach to CDX preparation with well-coordinated multiple short iterations should significantly shorten this key phase.

Phase 2: Dry Run

Organizing a CDX is like organizing a mission to the Moon. Every part of the complex infrastructure and all plans have to be well designed and tested before the start. The dry run is similar to beta testing a spacecraft without crew. It follows the same schedule and timing as final exercise to rehearse the entire scenario and interaction between *red, white* and *green teams*.

The testing is performed in the same infrastructure that will be used for the final exercise, but without real users (prospective members of *blue teams*). Instead, different people are invited to deputize *blue teams*.

A dry run is conducted even if existing CDX is repeated without changes. It is because cyber range resources are allocated temporarily only for the duration of the exercise and then it is necessary to test it again:

- **Milestone:** The cyber range infrastructure is completely tested and functional. Possible technical issues are fixed. Scenarios of red and white teams are finalized and orchestrated. Scoring and assessment of blue teams are adjusted.
- **Identified troubles:** Although the dry run follows the final training scenario, it takes a much longer time than the real training session due to the reparation of frequent errors and logical inconsistencies.
- **Solution:** Dry run cannot be omitted as cyber ranges are too complex, and a CDX represents an event with "the single attempt" when everything has to be working. The reduction of the cost requires the reduction of the frequency of errors so that the dry run could be restricted to only technical testing of unreliable infrastructure. Continuous testing and delivery introduced into the previous *preparation* phase can help to reach this goal. Using the plan-driven CDX life cycle can help. Formalization of artifacts and planning their delivery should enable us to use systems of automated deployment, e.g., Ansible (Hall, 2013). Also, unit testing can be introduced, which is completely missing in current ad-hoc CDX development. All these steps could make the beta testing substantially less demanding.

Phase 3: Execution

This phase represents the CDX event when real *blue teams* familiarize with the entrusted critical infrastructure, and then they defend it against activities of the *red team*. Simultaneously, they respond to requests of the *white team*.

A lot of run-time data is collected during this phase. The data captures activities of all teams, received penalties, etc.:

- **Milestone:** The CDX event was realized. Exercise data was collected for further analysis. Hardware resources were released.
- **Identified troubles:** A lot of organizing participants (members of *red, green*, and *white teams*) are necessary to organize a single CDX event.
- **Solution:** Automation of tasks. There are attempts to replace the interaction of real people with automated algorithms that are able to follow the training scenario and fulfill the tasks of red and *white teams*. The application of either an agile or disciplined approach to the CDX life cycle does not affect this phase.

Phase 4: Evaluation

During the exercise, all participants fully concentrate on their tasks. Especially *blue teams* have only limited awareness of what the *red team* is doing or what were the possible correct reactions to attacks or requests. Therefore, the primary goal of the evaluation phase is to provide feedback to *blue teams* so that they can learn from the exercise. Apart from that, the secondary goal is to retrospectively validate CDX and verify how much it fulfilled expectations of *stakeholders* and *training experts*. In both cases, the run-time data and notes of participants are collected, analyzed, and processed to feedback reports and internal lessons learned:

- **Milestone:** A feedback was prepared and delivered to *blue team* members. Internal lessons learned were formulated and provided to *stakeholders* and *training experts*.
- **Identified troubles:** Nowadays, it takes several weeks to collect and analyze necessary data and to prepare reports and other outputs. It is because the outputs are created informally and ad-hoc. Organizers of a CDX put together their notes, analyze collected data, and together produce desired feedback and internal lessons learned. A lot of manual analysis performed by domain experts is necessary.
- **Solution:** Evaluation is a creative process where people with different expertise have to collaborate tightly. People involved in this process are known in advance. They are *stakeholders, training experts*, and members of the *red, white*, and *green teams*. Considering these facts, the *evaluation* phase shows the signs of agile development.

On the other hand, the scope of their work is known (i.e., feedback reports and lessons learned), while the time required to prepare the outputs is flexible. Although we attempt to shorten the evaluation and provide feedback as soon as possible, we are aware that the quality of outputs depends on the quality of post-training analysis, which is time demanding. These aspects indicate that introducing a disciplined methodology would be more beneficial.

The traditional triangle *features/scope – resources/cost – schedule/time* used to distinguish between fixed and variable features of methodologies fails, indicating that a balanced approach should be considered. Information gathering should be based on a disciplined approach with formalized artifacts and processes. This formalization enables us to develop supporting tools that would shorten and precise data collection. On the other hand, subsequent agile, iterative creation of feedback and internal lessons learned would support orchestration of involved experts leading in faster outputs.

Summary

Figure 1 summarizes the application of agile or disciplined approaches to individual phases of the CDX life cycle. Using an agile approach to the *preparation* with short iterations, orchestration of people, and continuous testing and delivery of outcomes could significantly shorten this phase and reduce the *dry run* as well. On the contrary, the *evaluation* requires a balanced approach with disciplined information gathering and agile information processing. Neither a disciplined or agile approach has a direct impact on the *execution phase*.

DETAILED DISCUSSION OF DISCIPLINES

Disciplines in the software development process represent cross-cutting activities spread over all phases of the life cycle with variable intensity. Since the goal of the CDX development is not related to a cyber range but its content, the five disciplines discussed in this paper lightly differ from what is usually introduced in standard software development.

The goal of this section is to provide a fine-grained view of the character of activities so that the previous observations made during the analysis of phases are proved and explained in more detail. The text focuses primarily on the *preparation* and *evaluation* that appeared to be are relevant for the discussion on the usage of disciplined or agile approaches.

This section is structured as follows. For each discipline, artifacts that represent key tangible outputs are discussed. Then, the responsibilities of individual roles dealing with the artifacts are described, reveal the character of working activities. The approximate work effort required to be spent in various phases is suggested in Figure 1 in the form of bar charts and discussed for each discipline as well. Based on these details, conclusions regarding using either disciplined or agile approaches at the low level of CDX development are formulated.

Artifacts, roles, and responsibilities are also schematically captured in low-level models (see Figure 3, for instance) with the following notation: Responsibilities for the creation of artifacts are

captured by horizontal swimlanes with a list of involved roles on the top of each swimlane. For the sake of simplicity, activities are omitted. They are only discussed in the text. Instead, dashed arrows are used, representing dependencies (impacted by relationships of the SPEM meta-model) between artifacts. Arrows direct from a source artifact (a source of knowledge) to a target artifact (derived knowledge or specification). Artifacts produced by other disciplines are placed out of swimlanes and depicted with less intensive light gray color.

Business Modeling

Business modeling is optional in traditional software-system development. Its goal is to get insight into the business processes of the application domain that should be reflected in the implementation. Often, the business vision and objectives are formulated much earlier than the project is initiated.

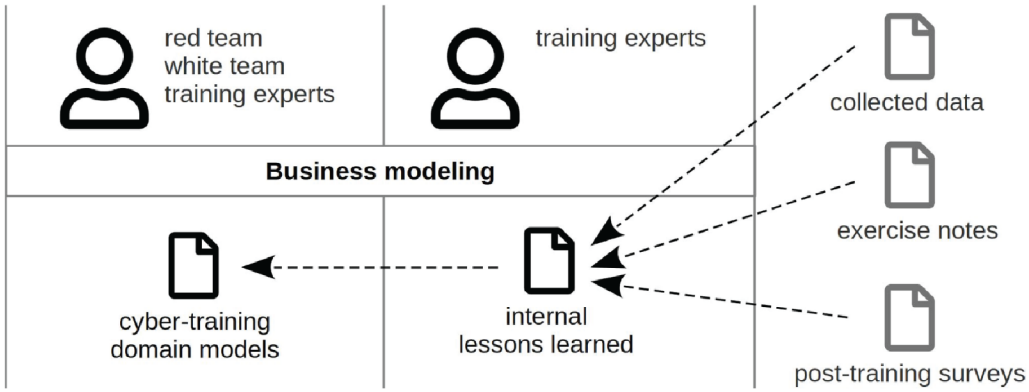
In the application domain of this paper, the business is related to hands-on cybersecurity training provided as a service. The business modeling, therefore, corresponds to the knowledge modeling in the field of learning and cyber security. The business view should cover two primary business goals.

First, it is a learning impact. Learning objectives can be derived from the analysis and modeling of existing cybersecurity processes, e.g., attack or cyber-defense strategies (Simmons, Ellis, Shiva, Dasgupta, & Wu, 2014; Mavroeidis & Bromander, 2017), so that they reflect new trends and threats.

Second, it is the sustainability of the training program. According to (Ryan & O'Connor, 2013), tacit knowledge of domain experts is acquired and shared directly through good quality social interactions and through the development of a transactive memory system. However, CDXs are organized occasionally, and the knowledge gained during the organization of a CDX is lost as people leave the development team. Methods of formal knowledge modeling (Bimba et al., 2016) have to be employed to support long-term knowledge sharing and transfer. Conceptual ideas of knowledge modeling in CDXs can be found in (Ošlejšek, Vykopal, Burská, & Rusňák, 2018), but further research is required in this field. To the best of knowledge of the authors, such pre-training analyses are not conducted in practice due to the missing methodology for CDX development even though they would significantly accelerate exercise preparation.

- **Artifacts, roles, and responsibility:** The involvement of user roles in the creation of artifacts and artifacts' dependencies are schematically captured in Figure 3:
  - *Cyber-training domain models:* currently, they have the form of informal text documents shared as wiki notes or, more often, they do not exist at all. Most of the knowledge related to the design of the content of CDXs keeps in heads of involved cybersecurity experts, lawyers, and legal experts invited to red and *white teams*. If formal modeling is introduced in the CDX life cycle, then the *red team* should be responsible for modeling cybersecurity processes, e.g., new vulnerabilities or attack vectors. The *white team* should contribute soft skills to the model, e.g., a classification of low-related objectives. *Training experts* should review the models to be applicable in the educational context.
  - *Internal lessons learned:* Experience gained from particular exercise and used as supporting material for future exercises and further development of the cyber range. Lessons learned are formulated by *training experts* retrospectively based on the analysis of the *collected data*, *exercise notes*, and *post-exercise surveys* provided by different people involved in the exercise, as discussed later in section *Deployment and Operation*. Lessons learned from individual exercises should also be retroactively reflected in the existing *cyber-training domain models*.
- **Work effort:** Business modeling is the most intensive at the beginning of the *preparation* phase, when learning and training objectives are formulated, and during the *evaluation* when lessons learned are derived, and business models are updated according to gained experience.
- **Disciplined vs. agile character:** Elaboration on the *cyber-training domain models* is significantly creative work requiring the collaboration and synchronization of many experts. Therefore, the agile approach in both the *preparation* and *evaluation* phases should be preferred. On the other

Figure 3. Dependencies between artifacts of the business modeling discipline and roles participating in their creation



hand, the formulation of *internal lessons learned* during the *evaluation* requires information structuring, formalization, and well-driven delivery of supporting materials. Otherwise, the outputs are either incomplete or hard to re-use for future exercises. Therefore, a disciplined approach should be preferred in this case. These observations confirm the agile character of the *preparation* phase and the balanced character of the *evaluation* phase.

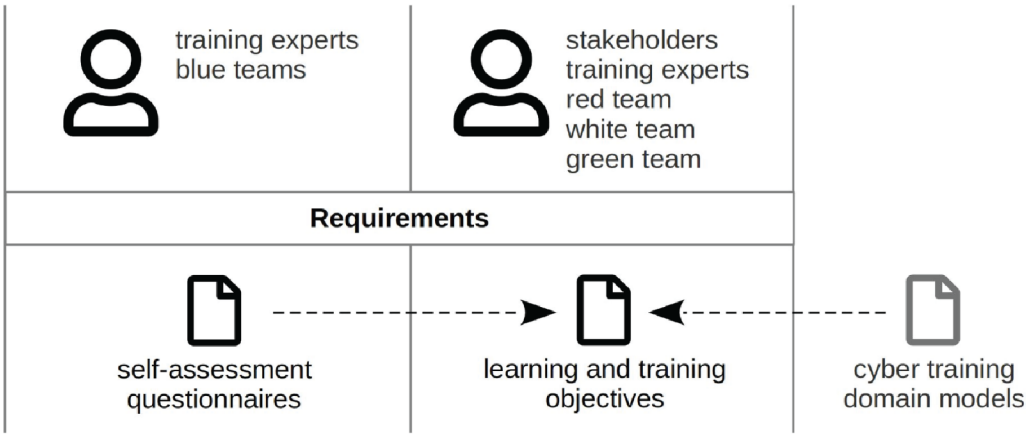
Requirements

Software development distinguishes two types of requirements: functional and non-functional. However, this traditional division fails in the CDX life cycle. Modern cyber ranges are designed as generic, enabling users to organize a wide variety of different exercises through a generic user interface. They are equipped with generic scoring boards, analytical tools, and interfaces providing access to hosts of the defended network, for instance. It is possible to use again the parallel with the ERP system providing a unified interface for variable business goals. Therefore, functional and non-functional requirements can be considered as fixed in this sense. The CDX development methodology deals with exercise development, not cyber range development.

Therefore, the CDX development distinguishes another two requirements: scenario- and infrastructure-related. *Scenario-related requirements* describe the activities of users involved in the exercise. They define what and when the *blue, red*, and *white teams* do in the cyber range during the exercise. On the contrary, *infrastructure-related requirements* are linked to the facilities of the cyber range. They include requests put on the configuration of the cyber range, e.g., minimal throughput of network connection.

- **Artifacts, roles, and responsibility:** The involvement of user roles in the creation of artifacts and artifacts' dependencies are schematically captured in Figure 4:
  - *Self-assessment questionnaires:* Part of prerequisite testing (Švábenský & Vykopal, 2018) of *blue team* members. Questionnaires provide insight into the experience and skills of individual trainees. They are defined by *training experts*. Results of self-assessment are used to adjust learning and *training objectives* and for establishing balanced teams.
  - *Learning and training objectives:* They define educational requirements that should fit the skills of trainees (*blue team* members) and expectations of *stakeholders*. They are defined by *training experts* together with *stakeholders* and they reflect *self-assessment questionnaires* and *cyber-training domain models*, if available. This artifact includes soft learning objectives as well as requirements put on network topology. *Red team* and *white team* members act as domain experts consulting and reviewing meaningfulness of the objectives from the cyber security and legislation points of view. The *green team* reviews network requirements from

Figure 4. Dependencies between artifacts of the requirements discipline and roles participating in their creation



the point of view of technical possibilities of the cyber range infrastructure. Learning and training objectives can be considered as critical because they form the basis for other artifacts. Improperly selected objectives can lower the impact of the exercise, demotivate trainees to finish the exercise, or demotivate *stakeholders* to further support the training program.

- **Work effort:** Initial requirements are specified during the early stages of the *preparation* phase and then adjusted continuously during this phase. Significant revisions are usually triggered by acceptance tests performed in the *preparation* and *dry run* (see section *Testing* for more details). *Self-assessment questionnaires* are usually taken once during the *preparation* phase. However, iterative prerequisite testing would be possible as well.
- **Disciplined vs. agile character:** As *stakeholders* require to train users in new skills, often related to their real critical infrastructures that they operate, CDXs are usually designed from scratch. The *learning and training objectives* that are key in this discipline have to be invented and defined from the beginning. Their elaboration requires long discussion between *stakeholders* and organizers with short iterations to reach initial definitions as soon as possible. These observations correspond to the agile character of the whole *preparation* phase.

Design

The ultimate goal of this process is to think over the details of the exercise, including technical specification being used for the configuration and initialization of the cyber range.

- **Artifacts, roles, and responsibility:** The involvement of user roles in the creation of artifacts and artifacts' dependencies are schematically captured in Figure 5:
  - *Scenario tasks and injects:* Attack plans of the *red team* and tasks of the *white team* (in the cybersecurity domain, tasks of the *white team* are called injects). Tasks and injects are derived from *learning and training objectives* with respect to the results of *self-assessment questionnaires* and the domain knowledge captured by the *cyber-training domain models*, if available. Since the scenario tasks and injects artifact is directly linked to the *learning and training objectives*, then also the participating roles are very similar. However, in this process *stakeholders* and *training experts* act as consultants checking whether tasks and injects proposed by *red* and *white team* fits learning and training objectives.
  - *Background story:* Aa fictitious story formulated using the fantasy of *stakeholders* and *training experts* and proving a broader context to *blue teams*. The story explains who is who in the cyber warfare, what is the organization whose network to be protected, what is the

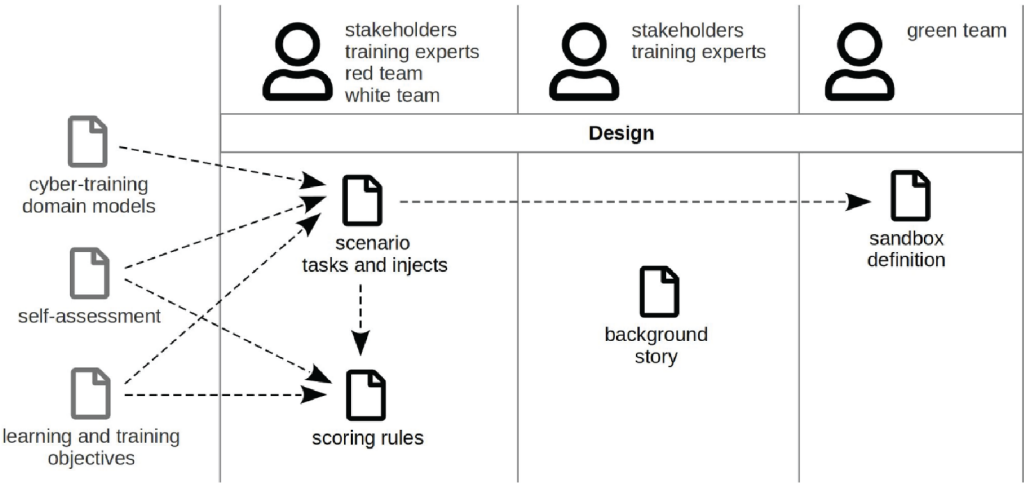
critical infrastructure, and other facts that help *blue teams* understand their goals. Fictitious countries in an escalating conflict are often used to provide trainees with a pseudo-realistic world fallen in the cyber warfare, where a critical infrastructure, e.g., nuclear power plant, has to be protected. This story is later transformed into information sources available to *blue teams* during the exercise, e.g., a news portal, information panels, or oral communication between the *white* and *blue teams*.
    - *Sandbox definition:* A structured document capturing the network topology. This technical document is built by the *green team*. It encodes parameters of links and hosts, e.g., throughput, amount of RAM, CPU speed, or IP addresses. It also defines software running on individual hosts. Besides the operating system and applications, it also specifies vulnerabilities that have to be presented on hosts according to the *scenario tasks and injects* artifact. Software to be running on hosts is prepared in the form of disk images that are uploaded on hosts during deployment.
    - *Scoring rules:* Penalties for unavailability of services, successful attacks of the *red team*, lax or unprofessional response to the requests of the *white team*, technical help of the *green team*, and other possible failures of *blue teams*. Scoring rules are often linked with specific scenario tasks and injects. *Scoring rules* are primarily defined by *stakeholders* and *training experts* who the best understand training and learning objectives. The *red* and *white teams* bring an insight into the difficulty of tasks and injects.
- **Work Effort:** During the early stages of the *preparation* phase, a significant effort has to be made to draft tasks, injects, and the background story based on the gradual clarification of learning and training objectives. Another important milestone is a hackathon (see section *Testing*) during which all the artifacts are finalized and prepared for the first acceptation testing. Artifacts of the design discipline are continuously adjusted after the hackathon and during the acceptance testing.
- **Disciplined vs. Agile Character:** All the artifacts defined in this discipline are complex and mutually connected. Their concurrent iterative development, together with the artifacts of the *requirements* discipline, is a must. Therefore, the agile approach to their elaboration during the *preparation* phase should be preferred.

Deployment and Operation

In this discipline, organizers configure the cyber range, operate it, and allocate resources. Laboriousness depends on the properties of the cyber range. But in general, these activities include a lot of continuous manual work.

- **Artifacts, Roles, and Responsibility:** The involvement of user roles in the creation of artifacts and artifacts' dependencies are schematically captured in Figure 6:
    - *Allocated sandboxes:* An allocated network infrastructure with respect to the sandbox definition. The infrastructure can be either emulated in a virtual environment (e.g., in a cloud) or physically wired. The first approach is common in modern cyber ranges. Regardless of the realization, it is always a lengthy, unreliable process. The experience of the researchers shows that even a cloud-based emulation takes long minutes or hours to allocate complex networks of CDXs. Moreover, the allocation often fails for various technical reasons. Manual intervention and continuous testing by members of *the green team* are, therefore, always necessary.
    - *Initiated cyber range:* Instantiated and properly configured cyber range connected with *allocated sandboxes*. Cyber ranges represent complex software systems consisting of many mutually cooperating components that have to be properly configured and orchestrated. Typical sub-systems that have to be initiated are online user tools, scoring, data monitoring, automated attack generators, traffic generators, etc. The configuration process follows

Figure 5. Dependencies between artifacts of the design discipline and roles participating in their creation



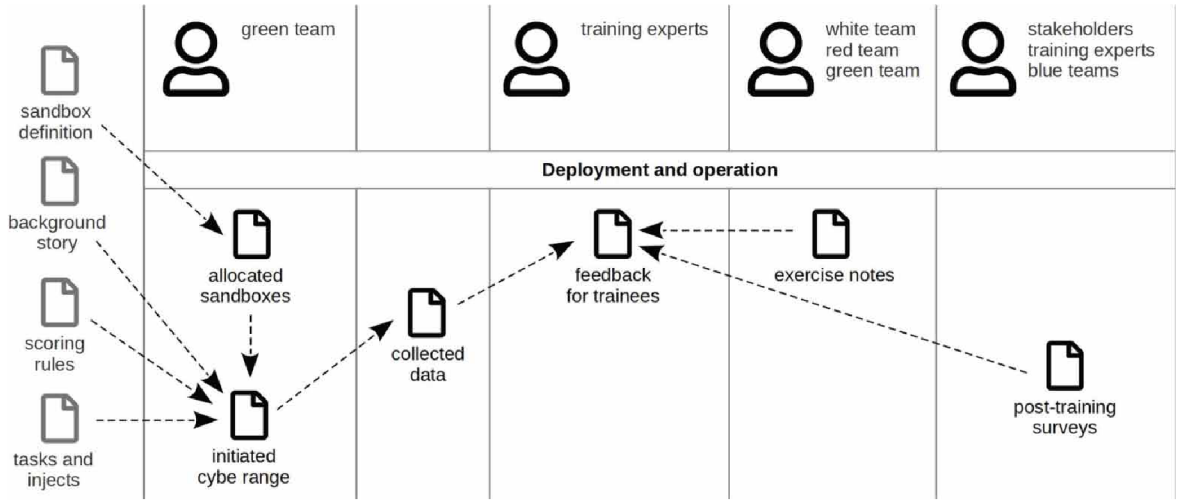
information included in the *background story, scoring rules*, and *tasks and injects*. The *green team* is responsible for the cyber range initiation.

- *Collected data:* A run-time data collected during the cyber range operation. The data is monitored and stored automatically by the *initiated cyber range*. The data captured during the *execution* phase and used for detail analysis of the exercise includes, for instance, performed attacks, injects, and their results, command history from individual hosts, or score development.
- *Exercise notes:* Experience of *red, white*, and *green teams* gained during the exercise.
- *Post-training surveys:* Questionnaires capturing the experience of *blue team* members. Surveys are defined by *stakeholders* and *training experts* to reflect their views on learning interests.
- *Feedback for trainees:* Results of the analysis of the *collected data* and personal experience of participants. Feedback enables members of the *blue teams* to learn from their behavior and mistakes. It has the form of statistical graphs, notes of *red, white*, and *green teams*, and other more or less formal explanations. Feedback is either created manually by *training experts* during the *evaluation* phase or automatically at the end of the *execution* phase.

- **Work effort:** Deployment and configuration activities are the most intensive at the end of the *preparation* phase, and right after the *dry run* when the cyber range is often reconfigured, and sandboxes are reallocated. Operational activities are dominant during the *execution* and *evaluation* phases when the data is collected and analyzed. The cyber range initiation, allocation of resources (sandboxes), and data collection can be significantly automated. The level of automation and continuous delivery is affected by features and possibilities of used cyber range.
- **Disciplined vs. agile character:** As the activities performed during the *preparation* phase include automated processes (cyber range initiation and the allocation of resources), the discussion of the application of either disciplined or agile approaches is irrelevant.

If the gathering of *exercise notes* and *post-training surveys* during the *evaluation* phase is informal, then agile preparation of the *feedback for trainees* would be used to deliver relevant information in a reasonable time. On the other hand, if the gathering of these artifacts is disciplined with predefined structure and deadlines, then the preparation of the feedback would be the matter of fast one-shot analysis. However, structuring the data is not that simple. It is possible to derive and classify common features of cybersecurity exercises, but the content and the realization of exercises differ. Therefore, it is necessary to support the gathering of unexpected informal pieces of information because they

Figure 6. Dependencies between artifacts of the deployment and operation discipline and roles participating in their creation



often provide very relevant and valuable pieces of information. A balanced approach to feedback preparation is, therefore, required.

Testing

Although the primary concerns of any CDX are related to learning impact, it is virtually impossible to test learning objectives and exercise difficulty. Organizers cannot reveal the content of the exercise to real trainees in advance to check its features. And tests conducted with dummy trainees are confusing due to their different skills and experience. Therefore, testing is restricted only to the verification of technical aspects and logical consistency of tasks and injects. In current practice, it is organized as two separate events dealing with two levels of acceptance testing.

A *hackathon* is equivalent to alpha testing. *Scenario tasks and injects* and *sandbox definitions* are evaluated by organizers in an intensive full-day workshop at the end of the *preparation* phase.

The *dry run* is equivalent to beta testing. Its goal is to verify the proposed cyber exercise completely and to get diverse feedback on it. The training session is conducted with dummy users, and then also this test can verify only technical aspects of the exercise, not educational. Since the *dry run* represents a separate phase that has been already discussed, it is omitted from further discussion in this section.

- **Artifacts, roles, and responsibility:** *Hackathon* is organized by *red, white*, and *green teams*. *Dry run*, in addition, involves *blue teams* but consisting of dummy trainees. During the acceptance testing sessions, observed flows are immediately repaired by revising artifacts discussed above. No new artifacts are created.
- **Work effort:** *Hackathon* is organized at the end of the *preparation* phase, followed by a short period of quick fixes of discovered errors. *Dry run* is in the CDX process model captured by a separate intensive phase.
- **Disciplined vs. agile character:** As the current practice in CDX testing is concentrated on two special events, these events increase time and cost. The best practices of agile development require ensuring the quality of the software product throughout the development process. Techniques of continuous testing and deployment are used to test early and often inside short iterations. Therefore, the already discussed agile approach to the *preparation* phase could reduce alpha testing and possibly eliminate the *hackathon*.

SUMMARY AND LESSONS LEARNED

This section summarizes observations made on the application of either agile, disciplined, or balanced approaches to CDX development.

CDX Life Cycle is Plan-Driven

The analysis of existing CDX life cycles revealed strong evidence of the plan-driven approach, similar to the RUP process framework, for instance. The life cycles consist of several well-defined phases, each specifying exact milestones, responsibilities, and artifacts. However, artifacts and development processes are often informal and ad-hoc in current practice. To introduce a real plan-driven methodology, their formalization and putting stress on their precise documentation is necessary. It can bring many benefits. Well-documented artifacts can be re-used in future CDXs. If they are well-structured, then they can also be used for the automation of selected processes. For example, the cyber range would be able to allocate complex sandboxes without the manual intervention of technicians automatically. Plan-driven development also brings better planning and management with verifiable deadlines and outputs. All these features contribute to the acceleration of the organization of CDX programs and their cost reduction.

The proposed unified CDX development method, which is based on the analysis of existing CDX life cycles, introduces four phases. Analysis of these phases revealed further details about their features that are summarized in Table 1 and discussed in what follows:

- **The *preparation* phase is agile:** The *preparation* phase shows the signs of agile development. This observation was proved by the detailed analysis of individual disciplines comprising of business modeling, requirements analysis, design, deployment & operation, and testing. Except for the deployment & operation, which turned out to be irrelevant, the application of agile approaches to other disciplines could significantly reduce the time and cost of this phase.
- **The *evaluation* phase is balanced:** The analysis revealed that the relevant disciplines of the *evaluation* phase are business modeling and deployment & operation. They show signs of both agile and disciplined features, making a balanced approach best suitable for the optimization of this phase.
- **The *dry run* and *execution* phases are not relevant:** Applying either agile or disciplined approaches to these phases does not make sense due to the nature of corresponding activities. However, their cost can be reduced by the already discussed introduction of the plan-driven methodology into the whole CDX life cycle and agile approach to the preparation phase.

Table 1. Identified characteristics of individual disciplines; n/a = not applicable

	Preparation	Evaluation
business modeling	agile	balanced
requirements	agile	n/a
design	agile	n/a
deploy and operation	n/a	balanced
testing	agile	n/a

CONCLUSION AND FUTURE WORK

Hands-on cyber defense exercises are crucial in educating the future workforce. However, their preparation is complex, then lengthy, and expensive. This research utilized standard methods of project management to analyze existing CDX life cycles and to derive its unified model. The proposed method shows that CDX development has a hybrid character combining both agile and disciplined features that have to be balanced. While introducing elements of agile development could improve the *preparation* and *dry run* phases, a balanced approach is required for the *evaluation*. Moreover, the whole life cycle is significantly plan-driven.

The main limitation of the presented research is its conceptual level of results. This paper provides a high-level view and generic discussion. The authors believe that even the gradual adoption of recommendations based on the observations presented in this paper can significantly reduce the cost of CDX preparation, making this kind of cybersecurity training more sustainable, available, and efficient. However, additional research elaborating on how the adoption should be realized in detail is necessary. The introduced CDX development method, together with observations made from the model, can serve as a framework for such investigation.

ACKNOWLEDGMENT

This research was supported by ERDF "CyberSecurity, CyberCrime and Critical Information Infrastructures Center of Excellence" project granted by the Ministry of Education, Youth and Sports of the Czech Republic [grant number CZ.02.1.01/0.0/0.0/16_019/0000822].

REFERENCES

Alliance, H. (2015). *CyberRX 2.0 Level I Playbook Participant and Facilitator Guide* (Tech. Rep.). HITRUST Alliance, LLC.

Bar-Yam, Y. (2003). When systems engineering fails - toward complex systems engineering. In *2003 IEEE International Conference on Systems, Man and Cybernetics (SMC'03)* (Vol. 2, pp. 2021–2028). doi:10.1109/ICSMC.2003.1244709

Baumgarten, G., Rosinger, M., Todino, A., & de Juan Marín, R. (2015). SPEM 2.0 as process baseline Meta-Model for the development and optimization of complex embedded systems. In *2015 IEEE International Symposium on Systems Engineering (ISSE)* (pp. 155–162). Rome, Italy: IEEE. doi:10.1109/SysEng.2015.7302749

Benzel, T. (2011). The science of cyber security experimentation: The DETER project. In *Proceedings of the 27th Annual Computer Security Applications Conference* (pp. 137–148). doi:10.1145/2076732.2076752

Beuran, R., Tang, D., Pham, C., Chinen, K.-i., Tan, Y., & Shinoda, Y. (2018). Integrated framework for hands-on cybersecurity training: CyTrONE. *Computers & Security*, *78*, 43–59. doi:10.1016/j.cose.2018.06.001

Bimba, A. T., Idris, N., Al-Hunaiyyan, A., Mahmud, R. B., Abdelaziz, A., Khan, S., & Chang, V. (2016). Towards knowledge modeling and manipulation technologies: A survey. *International Journal of Information Management*, *36*(6), 857–871. doi:10.1016/j.ijinfomgt.2016.05.022

Boehm, B., & Turner, R. (2003). *Balancing Agility and Discipline: A Guide for the Perplexed*. Addison-Wesley Professional.

Botta-Genoulaz, V., Millet, P.-A., & Grabot, B. (2005). A Survey on the Recent Research Literature on ERP Systems. *Computers in Industry*, *56*(6), 510–522. doi:10.1016/j.compind.2005.02.004

Brilingaite, A., Bukauskas, L., & Juozapavišius, A. (2019). A Framework for Competence Development and Assessment in Hybrid Cybersecurity Exercises. *Computers & Security*, 101607.

Davis, J., & Magrath, S. (2013). *A Survey of Cyber Ranges and Testbeds* (Tech. Rep.). DTIC Document.

Diaw, S., Cisse, M. L., & Bah, A. (2017). Using the SPEM 2.0 kind-based extension mechanism to define the SPEM4MDE metamodel. In *Proceedings of the International Conference on Computing for Engineering and Sciences* (pp. 63–69). New York, NY: ACM. doi:10.1145/3129186.3129199

Díez, E. G., Pereira, D. F., Merino, M. A. L., Suárez, H. R., & Juan, D. B. (2015). *Cyber Exercises Taxonomy* (Tech. Rep.). INCIBE. Retrieved from https://www.incibe.es/ extfrontinteco/img/File/intecocert/EstudiosInformes/incibe_cyberexercises_taxonomy.pdf

Dingsøyr, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). *A decade of agile methodologies: Towards explaining agile software development*. Elsevier.

Eagle, C. (2013). Computer Security Competitions: Expanding Educational Outcomes. *IEEE Security and Privacy*, *11*(4), 69–71. doi:10.1109/MSP.2013.83

EDURange. (n.d.). Retrieved 2017-05-22, from http://www.edurange.org

Ferguson, B., Tall, A., & Olsen, D. (2014). *National Cyber Range Overview. In 2014 IEEE Military Communications Conference*. IEEE.

Galván-Cruz, S., Mora, M., & O'Connor, R. (2017). A means-ends fesign of SCRUM+: an agile-disciplined balanced SCRUM enhanced with the ISO/IEC 29110 Standard. In *International Conference on Software Process Improvement* (pp. 13–23). Academic Press.

Ghayyur, S. A. K., Ahmed, S., Ali, M., Razzaq, A., Ahmed, N., & Naseem, A. (2018). A Systematic Literature Review of Success Factors and Barriers of Agile Software Development. *International Journal of Advanced Computer Science and Applications*, *9*(3).

Hall, D. (2013). *Ansible configuration management*. Packt Publishing Ltd.

Hoda, R., Salleh, N., & Grundy, J. (2018). The rise and evolution of agile software development. *IEEE Software*, *35*(5), 58–63. doi:10.1109/MS.2018.290111318

Imani, T., Nakano, M., & Anantatmula, V. (2017). Does a hybrid approach of agile and plan-driven methods work better for IT system development projects? *Development*, *1*(2), 3.

Kick, J. (2014). Cyber Exercise Playbook (Tech. Rep.). MITRE Corp.

Kruchten, P. (2004). *The rational unified process: an introduction*. Addison Wesley Professional.

Larman, C., & Basili, V. R. (2003). Iterative and Incremental Developments: A Brief istory. *Computer*, *36*(6), 47–56. doi:10.1109/MC.2003.1204375

Mavroeidis, V., & Bromander, S. (2017). Cyber Threat Intelligence Model: An Evaluation of Taxonomies, Sharing Standards, and Ontologies within Cyber Threat Intelligence. In *2017 European Intelligence and Security Informatics Conference (EISIC)* (pp. 91–98). Athens, Greece: IEEE. doi:10.1109/EISIC.2017.20

MCR. (n.d.). *The Michigan Cyber Range*. Retrieved 2017-05-22, from https://www.merit.edu/cyberrange/

Ministr, J., & Pitner, T. (2019). Towards Cybersecurity-Qualified Workforce. IDIMT-2019 Innovation and Transformation in a Digital World – 27th Interdisciplinary Information Management Talks.

Mirkovic, J., Benzel, T. V., Faber, T., Braden, R., Wroclawski, J. T., & Schwab, S. (2010). *The DETER Project*. Academic Press.

Misra, S., Kumar, V., Kumar, U., Fantazy, K., & Akhter, M. (2012). Agile software development practices: Evolution, principles, and criticisms. *International Journal of Quality & Reliability Management*, *29*(9), 972–980. doi:10.1108/02656711211272863

Object Management Group, Inc. (2008). *System & Software Process Engineering Metamodel (SPEM) 2.0* (Tech. Rep.). Retrieved from https:// www.omg.org/spec/SPEM/2.0/PDF

Ošlejšek, R., Vykopal, J., Burská, K., & Rusňák, V. (2018). *Evaluation of Cyber Defense Exercises Using Visual Analytics Process. In 2018 IEEE Frontiers in Education Conference (FIE'18)*. IEEE.

Petullo, W. M., Moses, K., Klimkowski, B., Hand, R., & Olson, K. (2016). The Use of Cyber-Defense Exercises in Undergraduate Computing Education. In *2016 USENIX Workshop on Advances in Security Education (ASE 16)*. Austin, TX: USENIX Association.

Pham, C., Tang, D., Chinen, K.-i., & Beuran, R. (2016). CyRIS: A cyber range instantiation system for facilitating security training. In *Proceedings of the Seventh Symposium on Information and Communication Technology* (pp. 251–258). New York, NY: ACM. doi:10.1145/3011077.3011087

Restuccia, D. (2015). *Job Market Intelligence: Cybersecurity Jobs* (Tech. Rep.). Burning Glass Tech. Retrieved from http://burning-glass.com/wp-content/uploads/Cybersecurity _Jobs_Report_2015.pdf

Rossey, L. (2015). *SimSpace cyber range*. Presented at the ACSAC 2015 Panel: Cyber Experimentation of the Future (CEF): Catalyzing a New Generation of Experimental Cybersecurity Research. Retrieved 2017-05-22, from https://www.acsac.org/2015/program/ACSAC%202015%20CEF%20Panel%20-%20Rossey.pdf

Ruiz-Rube, I., Dodero, J. M., Palomo-Duarte, M., Ruiz, M., & Gawn, D. (2013). Uses and applications of software & systems process engineering meta-model process models. A systematic mapping study. *Journal of Software: Evolution and Process*, *25*(9), 999–1025.

Ryan, S., & O'Connor, R. V. (2013). Acquiring and sharing tacit knowledge in software development teams: An empirical study. *Information and Software Technology*, *55*(9), 1614–1624. doi:10.1016/j.infsof.2013.02.013

Seker, E., & Ozbenli, H. H. (2018). The concept of cyber defence exercises (CDX): Planning, execution, evaluation. In *2018 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)* (pp. 1–9). Glasgow, UK: IEEE. doi:10.1109/CyberSecPODS.2018.8560673

Shuja, A. K., & Krebs, J. (2007). *IBM Rational unified process reference and certification guide: solution designer (RUP)*. Pearson Education.

Simmons, C., Ellis, C., Shiva, S., Dasgupta, D., & Wu, Q. (2014). AVOIDIT: A cyber attack taxonomy. In *9th Annual Symposium on Information Assurance (ASIA'14)* (pp. 2–12). Academic Press.

Švábenský, V., & Vykopal, J. (2018). Challenges Arising from Prerequisite Testing in Cybersecurity Games. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (pp. 56–61). New York, NY: ACM. doi:10.1145/3159450.3159454

Turk, D., France, R., & Rumpe, B. (2014). *Limitations of agile software processes.* arXiv preprint arXiv:1409.6600

Voříšek, J., Pour, J., & Buchalcevová, A. (2015). Management of business informatics model: Principles and practices. *Ekonomie a Management*, *3*(18), 160–173. doi:10.15240/tul/001/2015-3-014

Vykopal, J., Ošlejšek, R., Čeleda, P., Vizváry, M., & Tovarňák, D. (2017). KYPO Cyber Range: Design and Use Cases. In v. S. M. C. E. Cardoso J. Maciaszek L. (Ed.), *Proceedings of the 12th International Conference on Software Technologies* - volume 1*: ICSOFT* (p. 310-321). Madrid, Spain: SciTePress. doi:10.5220/0006428203100321

Vykopal, J., Vizváry, M., Ošlejšek, R., Čeleda, P., & Tovarňák, D. (2017). Lessons learned from complex hands-on defence exercises in a cyber range. In *Frontiers in Education Conference (FIE)* (pp. 1–8). Indianapolis, IN: IEEE. doi:10.1109/FIE.2017.8190713

Yamin, M. M., & Katt, B. (2018). Inefficiencies in cyber-security exercises life-cycle: A position paper. In *AAAI Fall Symposium: ALEC* (pp. 41–43). Academic Press.

Zaki, K. M., & Moawad, R. (2010). A hybrid disciplined agile software process model. In *2010 The 7th International Conference on Informatics and Systems (INFOS)* (pp. 1–8). Cairo, Egypt: IEEE.

Radek Ošlejšek received his Ph.D. degree in informatics from Masaryk University in Brno, the Czech Republic, in 2004 for the application of software engineering methods to the area of computer graphics. He is an assistant professor with the Faculty of Informatics, MU Brno. His current research interests include interactive visualizations, knowledge modeling, and exploratory data analysis in the domain of cybersecurity training. He participates in the research and development of the KYPO Cyber Range Platform.

Tomáš Pitner was awarded the Ph.D. at Masaryk University in 1998. Since 2008, he is an Associate Professor, Founder and Head of the Lasaris Research Laboratory, and an external Professor at the Faculty of Computer Science at the University of Vienna, Austria. Currently, he also leads a Research Program at the Czech Cybersecurity and Critical Information Infrastructure Center of Excellence (C4e). Since late 2018, he was nominated Secretary of International Advisory Board at the National Competence Center for Cybersecurity (NC3). His research focuses primarily on monitoring systems, critical infrastructures, namely power grids; web and enterprise software architectures and technologies. It also focuses on the communication aspects of academic and industrial cooperation. He leads large-scale R&D and contractual research projects in the field of power grids.