UNIVERSITY OF ECONOMICS - VARNA

FACULTY „INFORMATICS“

DEPARTMENT „INFORMATICS“

Svetoslav Stefanov Ivanov

Managing the Development, Maintenance and Support of Software from Technology Startup Companies

ABSTRACT

of a dissertation thesis for acquiring

of the educational and scientific degree "doctor" in professional direction 3.8. Economics, doctoral program "Application of computing technology in economics"

Research supervisor: Assoc. Prof. DSc Pavel Petrov

Varna

2022

The dissertation is 164 pages long, including 13 figures, 15 tables, appendices. The bibliography covers 140 literary sources.

The main results of the research were presented at scientific conferences and published in collections of reports (5 items), scientific journals (3 items).

The dissertation work was discussed and directed for defense before a scientific jury at a meeting of the "Informatics" department at the "Informatics" faculty of the University of Economics - Varna on 14.10.2022.

Scientific jury:

1. External members
   - Prof. Dr. Krasimir Todorov Shishmanov, SA-Svishtov
   - Assoc. Dr. Petya Emilova Popova, SA-Svishtov
   - Assoc. Dr. Rosen Ivanov Kirilov, UNSS-Sofia
2. Internal members
   - Prof. Dr. Vladimir Stoyanov Salov, UE-Varna
   - Prof. Dr. Yulian Andreev Vasilev, UE-Varna

The defense of the dissertation will take place on .... at .... hours in the hall .... of the University of Economics - Varna at a meeting of the Scientific Jury appointed by Order No. .... of the Rector of the University of Economics - Varna.

The defense materials are available to those interested on the website of the University of Economics - Varna, www.ue-varna.bg

I. GENERAL CHARACTERISTICS OF THE DISSERTATION

1. The importance of the research

As a result of the mass entry of information technologies into business and human life, prerequisites are created for the emergence of the phenomenon of "technological start-up companies". These companies represent a new organizational form, characterized by the initiation and development of new business, combined with high dynamics of processes in this direction. A part of the technological start-up companies focusses their activity on the production of software, where we can note that two key components are present - on the one hand, it is the software as an expected final product, on the other - the innovation process as an object of management.

In recent years, software has evolved from a complex and high-tech product intended for high-paying customers to a mainstream business and consumer product with increasing demands. In order to survive in the complex competitive environment, technology start-ups need to adapt their software production to these increasing demands. Modern software requirements include both providing desired functionality and meeting a number of quality and time-to-market criteria, as they are of key importance to the company's operations.

Small and medium-sized enterprises are considered the backbone of the economy and potential for growth, an important source of employment and a generator of new jobs. Startup companies are in most cases small and medium-sized enterprises that focus on innovation and rapid growth. The European Commission adopts a number of initiatives to improve the economic and regulatory frameworks for start-ups as a potential for economic growth and job creation. The process of creating and developing technology start-ups is associated with a high degree of uncertainty and risk. Risk for startups is often measured by the level of survival (or failure) beyond the fifth year of founding. According to some estimates, this indicator is below 20%.

The above regarding the important role of technology start-ups for the economy and at the same time their low survival rate, combined with the complexity and risk in software development shows **the relevance and importance** of the topic of managing the innovation process in software development in technology start-ups. Research is needed to support the development, maintenance, and software management activities of these types of companies.

Technology start-ups are characterized by relatively few resources, limited time window, limited entrepreneurial and marketing opportunities, high degree of uncertainty and risk of failure. In the beginning, not all the details regarding product, market and organization are clear. The production of a quality product can be achieved with the application of appropriate technologies, organization of processes and their effective management. The latter implies the use of a software system supporting management processes.

2. Research thesis

The main research thesis of the dissertation is that technology start-ups have a specific purpose, processes and organization, therefore entrepreneurs need a specialized management approach, combined with a software system, with the help of which they can achieve their goal. Of course, it should be considered that this is only one of the necessary conditions in the development of risky business ventures.

3. Goals and objectives of the study

The research objective of the dissertation is to develop a software system design that, applied with an appropriate process management approach, will successfully support the operations of technology start-up companies. In view of the set goal, the main tasks to be solved are:

1. Exploring the nature and characteristics of technology start-ups, as well as some issues related to software development.
2. Exploring software development and software project management approaches in technology start-ups.
3. Development of a project of a software system supporting software project management processes in technology start-up companies.
4. Presentation of a plan for the development and implementation of a software system.
5. Approbation of the research results in a specific object of application.

4. Object and subject of the research

The subject of the dissertation research are technological start-up companies engaged in the development, maintenance and support of software products. They may be entirely our own commercial products in a certain form and implementation (product, service, component, etc.) or by order from an external contractor.

The subject of the study is the development of new software products in an entrepreneurial process with unclear requirements and parameters in advance and dynamically changing over time. Therefore, we direct the research to appropriate approaches for managing the software development process and designing a software system that can support technology start-ups in this activity.

Taking into account, on the one hand, the fact that the topic thus set is comprehensive and another research approach can be applied to its study, and on the other hand, taking into account the set goals and objectives, it is necessary to define the following limitations:

1. Outside the scope of the study are questions related to the legal and regulatory frameworks to which the activity of the technology start-up company as a business entity (merchant within the meaning of the Commercial Law) must be subject.

2. The practical problems of the technology start-up company of a general nature for every company are not covered - relations with business partners, institutions, property management, capital, human resources, etc. In the dissertation, the focus is solely on software project management issues.

3. In a broader sense, the target activity of a technology start-up company may not be related to the production of software. Accordingly, these companies are not the subject of the study.

5. Research methodology

For the methodological basis of the study, various research methods were used in the dissertation, the most important of which are: research and data collection, comparison, analysis and synthesis, systematization (classification and typification), induction and deduction, modeling and scientific abstraction.

To describe the sequence of processes, a systematic approach to the researched object and subject was used. Visualization of various facts and data is carried out using the graphic, schematic and figural approach. The combined application of research methods and approaches is aimed at reaching final findings and presenting recommendations.

6. Approbation

Three articles in scientific journals and five reports in proceedings of scientific conferences have been published on the subject of the dissertation.

II. STRUCTURE OF THE DISSERTATION

The dissertation consists of an introduction, three chapters and a conclusion, and is 164 pages long, including 13 figures, 15 tables, appendices. The bibliography covers 140 literary sources. It also contains a list of abbreviations used.

Content:

III. SYNTHESIZED PRESENTATION OF THE DISSERTATION

Chapter 1. Theoretical foundations of software technology startups

As a result of the research in the first chapter, it is found that technology start-ups have a specific definition, purpose and need to develop both their own product and an adequate business model to prove viability and achieve growth. The choice of methodology, model and solutions should be assessed according to specific circumstances (problem, resources, entrepreneur's mission, product and market, risks). Combinations with a basic process from the listed and for specific elements - methods, tools and indicators from other approaches are possible, so that gradually with the development of an idea, business model and software product, the standard stages of business management are reached.

In the **first paragraph**, the nature and characteristics of technological start-up companies are presented.

It is known that a key role in the creation and functioning of technological start-up companies is played by the so-called entrepreneur. Its main activity is to organize work processes that are different in nature and structure in order to achieve a certain goal. Organizing, as a major branch of management theory, addresses issues related to the ways in which an organization can achieve its goals and objectives.

For the purposes of the dissertation, based on the criteria defined in this way, we can use the following working definition for the term **"technological start-up company"** - a newly created technological micro- or small enterprise[1], run by an entrepreneur[2] or an entrepreneurial team that has a short history (up to 5 years) with the primary goal of creating a new software product, effectively developing and validating a scalable business model for it, so as to prove

[1] We mean a micro- or small enterprise within the meaning of the SME Act and the European Commission's definition of an enterprise with up to 50 employees and a turnover/assets of up to 10 million euros.

[2] By "entrepreneur" we mean one or more entrepreneurs founding their company.

viability and ensure growth. Product development is related, directly or indirectly, to software development. The way to achieve this is through innovation in a product, technology, business model or organization, and discovering market opportunities.

The organizational legal form of a technology start-up company with an entrepreneur and a team developing a new software product and business model can vary according to circumstances. This is usually an independent new trader (within the meaning of the Trade Act) or a new business unit within an existing large company which has the same purpose. In the context of the conditions in Bulgaria with prevailing micro- and small software enterprises, it is difficult for the existing ones to start a new product with a team more than a micro-enterprise. Typically, start-ups are micro or small enterprises until they demonstrate significant growth where the organization itself grows.

In the literature there is an analogous term from the English language called startup or start-up. We assume that a start-up company and a startup are similar, having already specified that the criteria of innovation, vitality and growth are considered for start-up companies. Another feature is that startup companies are often designed to be subject to external financing and acquisition in order to accelerate growth or sell for profit. This topic is beyond the scope of this thesis.

As a feature of software development in a technology start-up company, we must also note the specificity of the activity itself. Basically, two important concepts in successful software development are user involvement and description and stability of requirements. For the tech start-up, these concepts don't apply, or they don't.

In the beginning, the user or customer may not be clear, the requirements may not be obvious, they may not be known, or they may change during the search for a niche market. Feedback from the user/contractor does not exist or is not always direct. It is usually not by directly asking questions and getting answers. It can be obtained by observation, research, data analysis, experiments to validate a hypothetical requirement and other methods. In addition, each requirement should also have an analysis of the marketing and financial effect, since its inclusion may or may not attract customers, but it certainly requires the commitment of resources for its development and implementation itself.

The state of entrepreneurship in Bulgaria in the IT sector, in comparison with other European countries, is developing at a pace corresponding to the business environment. Sofia has the most start-up companies.

In conclusion, the studies carried out reveal to a large extent the nature of technology start-ups from a legal and scientific point of view. In order to fulfill the research objective, it is necessary to identify the main software development problems that technology start-ups may face.

In the **second paragraph**, methodological problems related to the development of software products are investigated.

For the concept of "software product" we accept the definition of IEEE - a set of computer programs, procedures, rules and possibly accompanying documentation, as well as data related to the functioning of a computer. The business of the technology start-up company includes the specification, development, implementation, sale, support of software as well as services for it, but the emphasis is on production and support. The activity can be classified as:

– production and sale of own commercial products;

– own components for embedding in products of other developers;

– manufacturing by assignment from customers (external contractors);

– individual services for setting up, adapting and implementing a product;

– support of existing and implemented products from the above categories.

Each option has specifics in terms of assignee, task, funding, deliverable, life cycle, repeatability, etc., and a tech startup's business can be a mix of different options with different products. Each one can be implemented in one or a series of projects. Projects, as is well known, have scope, quality, cost and time as the main parameters, which are defined in a development contract with an external client or an in-house product development assignment. A general description of the options is given in table 1.

Table 1

Types of activities in a technology startup company (author's development)

Type of activity	**Contractor**	**The assignment is determined by**	**Result**	**Repeatability and renewal**	**Product specific activity**
Own products	R-l team / dept. "Marketing"	R-l team / dept. "Marketing"	Product ready for sale	Ongoing maintenance and av.	All
Proprietary components	R-l team / dept. "Marketing"	R-l team / dept. "Marketing"	Component for sale	Ongoing maintenance and av.	All
Client project	Customer	The client (or with a designer)	Custom software	None or with a new contract	Specification and support are options

Type of activity	Contractor	The assignment is determined by	Result	Repeatability and renewal	Product specific activity
Individual services	Customer	The client (or with a designer)	Embedded product	There is no	Specification, Documentation, Implementation
Support	Client / dept. "Sales"	Client / dept. "Marketing"	Service to specification	Urgent / as needed	Documentation, research, fixes

Note: It is possible to have separate "Marketing" and "Sales" departments with the main task of entering certain market niches. In case there are no separate departments, the team leader and the client perform their functions.

The software product for an external customer is on an assigned project (from direct contact, tender, upgrade of another developed product, experiment, others). Often, such projects are undertaken by technology startups in order to test an idea, gather product requirements, study a customer for subsequent repeatability and standardization. The parameters are defined in a contract before the actual work on the project. The scope and specification are set by the customer or prepared by an analyst at the technology startup.

Proprietary software products are a company initiative that has responsibility for planning, specification, implementation and funding. The requirements are from an internal contractor (r-l product, marketing specialists), but are supplemented in a different form with feedback from users. Quality and price (budget) are also determined by the company according to the chosen market positioning and capabilities.

A proprietary software component is analogous to a proprietary software product. A component is an intermediate product, a finished "part" included in software products. Their users are usually industry professionals (developers, programmers, designers, planners). It follows that the documentation and support are oriented towards professionals. The positive effect for the technology start-up company is selling to a specialized market and taking advantage of niche markets of other end product "producers". Details of the components are given below.

Individual services are a one-time action and service for a specific client (external contractor). These are beyond the scope of the present study. In some cases, where the service is more complex, they may be interpreted as work on an external client project. The difference is that we are talking about setup, implementation or integration services of an existing product (own, from an external project or from a third party).

Product maintenance is usually for contractually defined terms and obligations. It can be permanent with monitoring or support in the case of a problem reported by the customer. The distinctive thing is that work is regularly carried out on monitoring or complaints about defects. It may also include minimal product refinement.

Often, the software maintenance process includes the support process, but depending on the context of use of the term, there may be a difference between the two processes. According to international standards for the life cycle of software systems, maintenance includes activities to ensure the functioning of the system. When public procurement is announced in our country, "support" means support for users, i.e. training and assisting staff in learning the specifics of working with a new software system.

We can summarize that support is related to ensuring the normal operation of a built and implemented product, and support is related to making changes to correct identified errors, adaptation to a new environment, operating system, hardware, to improve existing ones and add new functional possibilities. Very often the importance of support is overlooked by IT professionals as they prefer to work on creating new software (based on personal experience).

Software development is the highly intellectual work of creating a new product. Unlike standard production, where the product is one-of-a-kind, here a completely new product is formed. The dynamics of requirements and technologies add additional complexity. Complexity management and product innovation can be achieved with the right team, process, environment and governance. In this sense, the importance of **human resource management** stands out. Some essential characteristics of human resources in a technology startup can be grouped into the following groups[3]:

– a team of professionals who effectively create an innovative and quality product;

– need for new knowledge, skills and qualities of specialists;

– a high share of human resources costs and a tendency for salary growth and mobility.

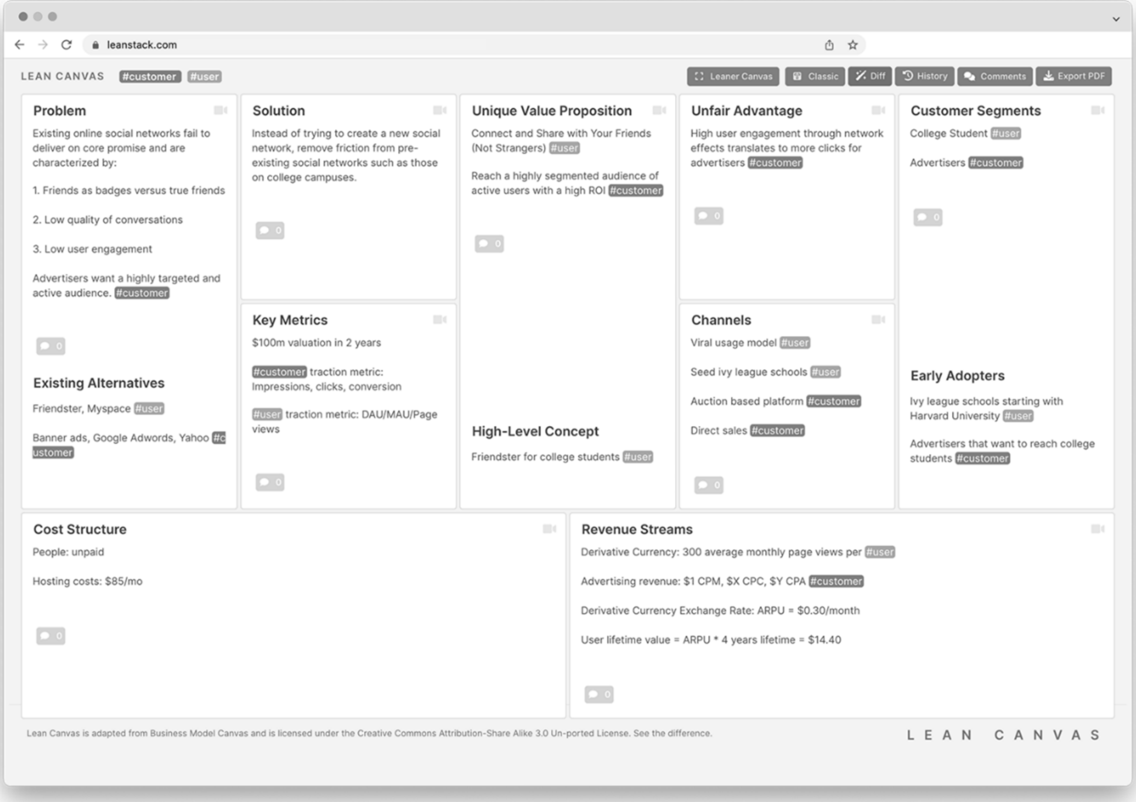
The place of *team motivation is important*. This is a wide-ranging process with many conditions (not just monetary incentives). The team leader must integrate knowledge and understanding of business needs, technologies and their use. We can summarize that human resource is the most complex

[3] Cited groups and other additional ones are defined in (DeMarco, Lister 2013)

element to manage, but the only one that can contribute to significant achievements in a technology startup company.

There are various **software development methodologies** suitable for startups. Of interest are the so-called "light methodologies", which are essentially simplified and with a low level of formalization and verification. An adaptation of the general approach "Lean Development" is of interest, which for the development of a start-up business is called "Lean Startup". In this approach, the basic concept is that, with certain principles and tools, the team should concentrate on delivering "value" (in the sense of satisfied valuable needs) to the user in the shortest possible time and with high quality, which is achieved through effective " value stream.

It is possible to use basic models to describe business models, business plans, but for a start-up it is important to concentrate on the essentials. From this point of view, methods and tools for describing models are described in the literature. For example, a tight-start scheme (see fig. 1) for business model development.



Problem

Existing online social networks fail to deliver on core promise and are characterized by:

1. Friends as badges versus true friends

2. Low quality of conversations

3. Low user engagement

Advertisers want a highly targeted and active audience. #customer

Existing Alternatives

Friendster, Myspace #user

Banner ads, Google Adwords, Yahoo #customer

Solution

Instead of trying to create a new social network, remove friction from pre-existing social networks such as those on college campuses.

Key Metrics

$100m valuation in 2 years

#customer traction metric: Impressions, clicks, conversion

#user traction metric: DAU/MAU/Page views

Unique Value Proposition

Connect and Share with Your Friends (Not Strangers) #user

Reach a highly segmented audience of active users with a high ROI #customer

High-Level Concept

Friendster for college students #user

Unfair Advantage

High user engagement through network effects translates to more clicks for advertisers #customer

Channels

Viral usage model #user

Seed ivy league schools #user

Auction based platform #customer

Direct sales #customer

Customer Segments

College Student #user

Advertisers #customer

Early Adopters

Ivy league schools starting with Harvard University #user

Advertisers that want to reach college students #customer

Cost Structure

People: unpaid

Hosting costs: $85/mo

Revenue Streams

Derivative Currency: 300 average monthly page views per #user

Advertising revenue: $1 CPM, $X CPC, $Y CPA #customer

Derivative Currency Exchange Rate: ARPU = $0.30/month

User lifetime value = ARPU * 4 years lifetime = $14.40

Lean Canvas is adapted from Business Model Canvas and is licensed under the Creative Commons Attribution-Share Alike 3.0 Un-ported License. See the difference.

LEAN CANVAS

Fig. 1. Lean Canvas for developing a startup business model. Source: Leanstack <https://leanstack.com/lean-canvas> (23.08.2022)

The Lean Canvas for Business Model Development for Startups has 9 related elements and is typically presented as a one-page outline. It is based on the Business Model Canvas created by Alex Osterwalder (https://www.alexosterwalder.com/). The main elements of the scheme are the following: problem (problem), solution (solution), the unique combination of products and services that has value for the customer (unique value proposition), unfair advantages (unfair advantage), customer segments (customer segments), key metrics (key metrics), distribution channels (channels), cost structure (cost structure), sources of income (revenue streams). An online tool for its creation is also available - https://leanstack.com/lean-canvas.

The **third paragraph** examines software project management in technology startups.

For a startup's business model to be successful, a management organization must be created to execute and improve it. Organizing is seen as a key concept in management theory. The created management organization can achieve a defined goal that is beyond the capabilities of one person through more productive methods - division of labor, more large-scale and modern technologies, cost savings, etc.

From this point of view, technology startups perform specific activities and through their results the goal is achieved. In turn, structure determines the behavior of technology start-ups, and it is of utmost importance because it creates commitment to common goals and values and has an impact in connecting employees with the external environment, as the output of the activity is manifested outside the organization.

For technology startups, decision-making is fundamentally about risk and uncertainty. Therefore, a more detailed definition of management functions can significantly improve work, as it is possible to avoid some problematic situations at an earlier stage, which is also more optimal in view of the overall functioning of the organization in the long term.

In software development, the methods of empiricism and the principle of feedback are widely used:

– processes move according to the "pull" principle (also known as Kanban);

– the manager is present at the place, among the employees, where the product is created;

– always aim to test and evaluate everything possible, including employee evaluation, which may not be appropriate for the initial stages of a startup due to lack of time and resources;

- strategic management is a process, not an analytical approach - the strategy is formed from an active and flexible internal environment, gradually, in interaction with the external one;

– experimental adaptation by learning from experience instead of advance long-term forecasting.

These principles build on classical management theories in a direction to work in smaller markets, more flexible product models, in a more volatile external environment. On this basis, the model for the learning organization is created - learning through experience and system groups and processes for improvement; benchmarking model - comparison with industry leader; double circle model – to receive feedback from partners and competitors about changes in the environment; network model – the external environment is not just the generalized market, but a complex network of interactions with many organizations in different dimensions; offering solutions based on small permanent groups, constant improvement.

Coordination practices that are frequently used in these approaches and are common in literature and practice are summarized in table 2.

Table 2
Summary of coordination practices described in theory and common in practice
(author's development)

Coordination practice	Description
Splitting into project-led small groups	The product is decomposed into modules or parts, each group dealing with one part
Splitting development into steps (iterations)	Each adds or modifies the product in part with all activities
Regular team sync meetings	Planning, daily work, end of stage or feedback project
Applying a Project Management Method	Tasks, resources, team and time distribution, incl. reports for team, step-by-step, personal performance, shared project documentation
Liaison role with "business outsourcer"	Introduction of a "business analyst" role with a requirement specification task
Liaison role "project manager"	Manages everything about the project using a chosen project management method
The connecting roles are in the team	Roles "business analyst", "project manager" and others are part of the team and participate in all team meetings and decisions
Coordinating decisions with options	When deciding on interdependent elements, there is always a group of options to choose from and stack on both sides
Synchronization of interfaces	Separation of the product into modules and teams, and department heads and architects coordinate the "connectivity of the modules" to be clear, stable and little changeable
Development of layers	Software is developed in layers so that the system is resistant to changes in one layer without the need for adjustments in another (module packaging)
Information radiators	Information about the status of a project/product is available to all, by visually displaying the progress in a visible and easily accessible place

Coordination practice	Description
Information boards	The most important operational reports are in a visible place for everyone according to the authority and updated constantly
Work in pairs	Two specialists work on one task
Work in one office one team	For direct horizontal communication
Code review, mutual approval	Primary quality control through peer approval and control of compliance with work standards
Regular/automated product integration	At each change or daily from ongoing work, the product is assembled ready for testing
Regular/automated testing systems	Regular feedback on product quality and need for corrections
Virtual office	Share information in the team regardless of the workplace
Technology groups and meetups	For the exchange of what is being worked on, how, what problems there are and technological exchange on a functional basis
Matrix structures	The specialist is in a functional department where he develops professionally, but mainly works in a project with a project manager
Following standard processes	The process is well defined and its implementation by all participants ensures mutual coordination
Open standards for technologies and solutions	Publicly available and popular standards, technologies, solutions, tools are used, in order to save development, easier integration of modules, training

The modern understanding of successful management affirms the need to build both good business processes and an appropriate and efficient software system. The software system helps technology start-ups to introduce new forms of management with higher efficiency and to optimize processes. It should provide information for timely, adequate and effective decision-making.

In the **fourth paragraph**, different approaches to software development management are discussed.

The following are examined in more detail: structured approaches, the object-oriented approach, agile approaches, the Evo approach (Evolutionary Value Delivery) - an evolutionary approach, the UP approach (Unified Process, unified process) and its commercial variant RUP, the SCRUM approach (Systematic Customer Resolution Unraveling Meeting) , the XP approach, the DSDM (Dynamic Systems Development Method) approach, the FDD (Feature-Driven Development) approach and the LSD (Lean Software Development) approach.

There are also many other agile approaches such as: "Highsmith Adaptive Software Development", suitable for large manufacturers of development environments (eg Microsoft); approaches for accelerated development, etc. "fused development"; "Development with prototypes" approach as an alternative to the structural/cascade model in order to reduce the effect of design errors; "Development with the user's participation" approach, through which, through a set of techniques and targeted actions, the active participation of the user is ensured in all phases of the process; "Development with multiplication" approach, which relies on the use of existing software elements (ideas, algorithms, models and most often software components) in new software products; "Reliable Software Development" approach to high-criticality software development where the goal is to minimize defects through detailed formal specifications, design with encapsulation (only necessary for the task), quality assurance mechanisms with systematic verification and validation, rigorous testing following them, watching for dangerous structures, etc.

For the needs of tech start-ups, only the more important approaches, which we believe are also more suitable for small teams, are discussed in more detail.

Chapter 2. A Software Production Management System in Technology Startups

The chapter presents the developed conceptual and logical model of a software system managing the main processes and activities in the development, maintenance and support of software. Small companies are characterized by a lower degree of formalization, simplified procedures and rules. Therefore, regarding the scope of the software system, we assume that software development covers a certain set of functionalities, avoiding great complexity and getting into purely technical issues regarding design, programming, testing and integration, document management, configurations, etc. Specialized software can be used for them at individual discretion.

In the development of the model, we limited ourselves only to the basic business scenario of interaction with the system, for which interaction diagrams and collaboration diagrams were developed to show how employees with different roles in the technology startup company interact and the scope of their responsibilities. Their responsibilities are the basis for shaping system functions and custom access for each role.

The logical model was developed based on the proposed conceptual business model, and for its implementation, the business entities were initially identified and developed as classes with corresponding properties and relationships.

In the **first paragraph**, a conceptual model of the software system is presented. The main business processes and activities of the system for development, maintenance and support of software, and the information business modeling of the system are given.

A software production management system in technology start-ups needs to cover all aspects of management where process status information is essential to achieving goals or legally binding. Small companies are characterized by a lower degree of formalization, simplified procedures and rules. For the purposes of our study, we accept the limitation that we consider technology start-up companies whose main activity is software product development and their support. As a result of the studies of the world experience in managing the software development processes made in the first chapter, we consider particularly appropriate the so-called "light methodologies" with an emphasis on implementation of the so-called "customer value product" when working with a small team and limited resources. Therefore, we believe that the project of a software system should be concentrated only on the main activity - production and maintenance of software products, as well as the most necessary details for it. The remaining parts of the overall software system of a technology start-up company can be implemented with a ready-made model or product at a relatively low cost and easy implementation.

A business scenario for interaction with the system (BSS) is shown in fig. 2, through graphical elements, according to the UML 2.5 specification. It describes the interaction between business actors and the software system in software development. Scenarios are represented by oval shapes. Relationships between actors and scenarios show the actions being performed. The link "uses" indicates that one script uses another, and "includes" indicates that one script includes as part of itself another script.

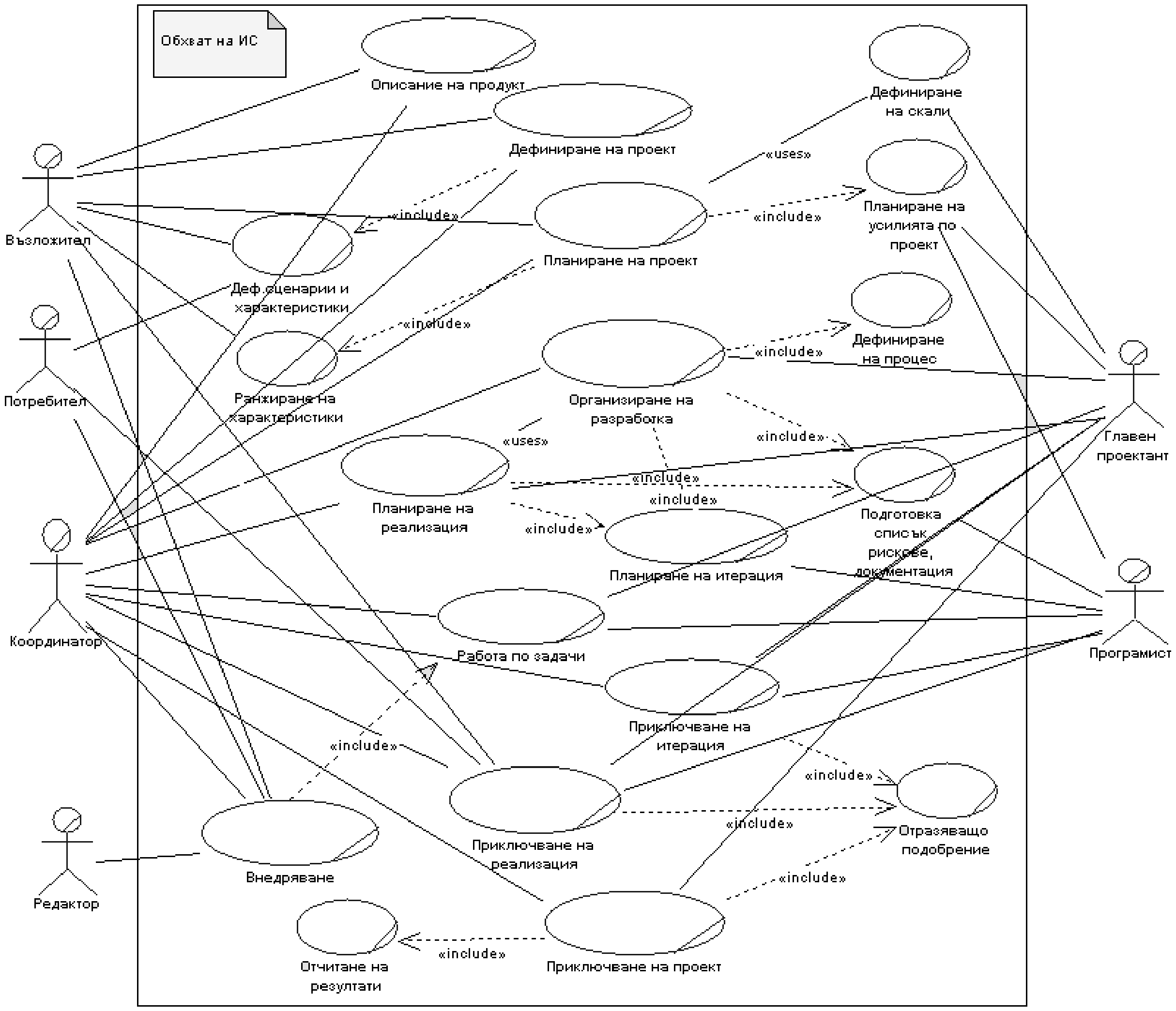
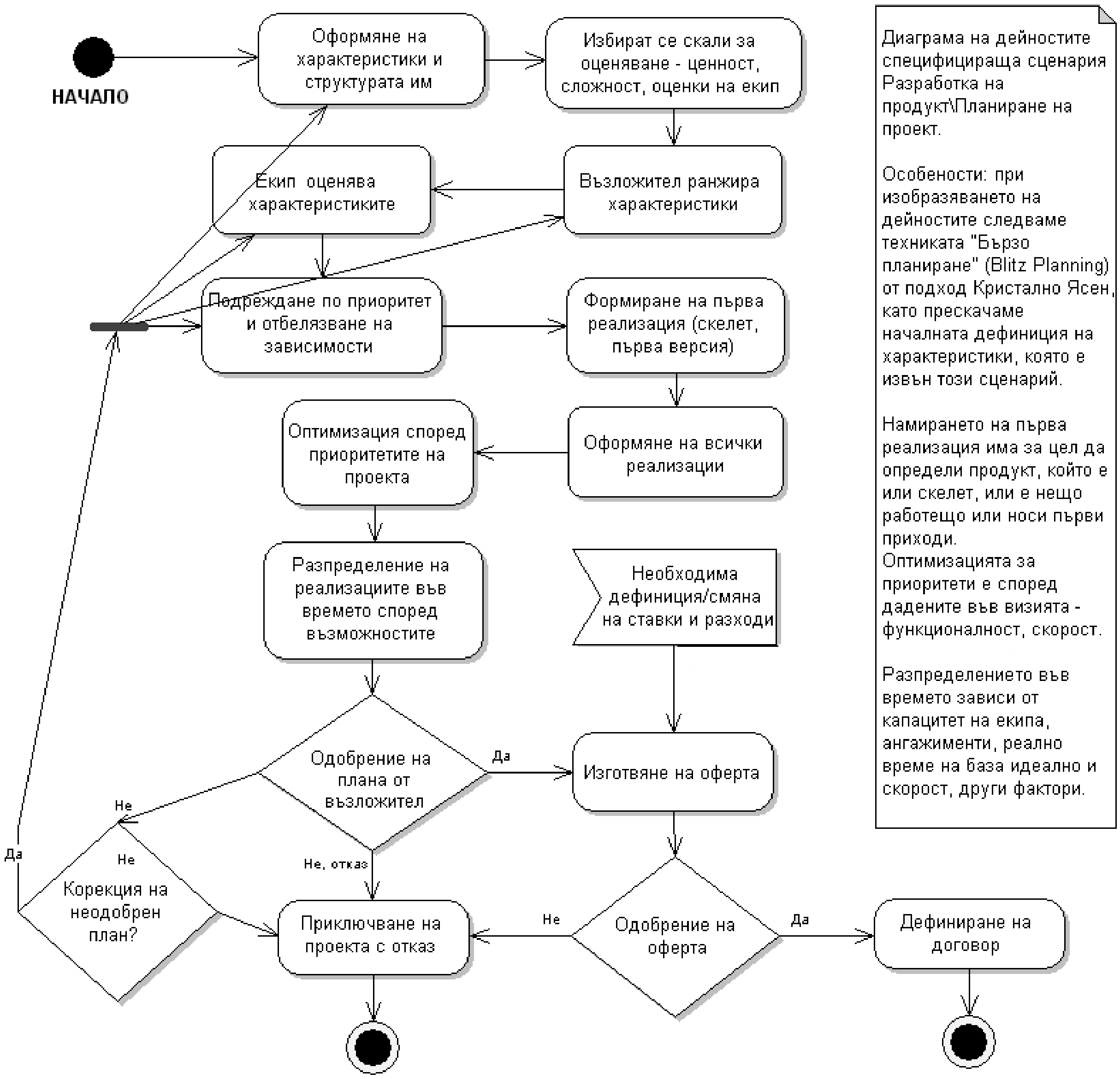


Fig. 2. Diagram of the main business scenario for interaction "Product development" (author's development)

Regarding the implementation of the BSVS, interaction (sequence) diagrams and collaboration diagrams were developed, which show the interaction between business workers (with roles in the technology start-up company - coordinator, chief designer, programmer, editor, tester) and their frameworks responsibilities. Their responsibilities are the basis for shaping system functions and custom access for each role. The business actors are the external participants – the client and the user.

In fig. 3 presents the details in individual business scenarios with an activity diagram and shows the activity description of a given scenario. The detailed representation of the business entities in the object model is given in the diagram in fig. 4. It also shows the connections between them. A state diagram is used to reflect the states of these business entities and the events that change them. Examples regarding the "design states" and "realization" components are given in fig. 5 and fig. 6 through a diagram of possible states of entity "Project" and the events in which these states can change. These diagrams are also the basis for creating the list of statuses for the given object, as well as the logic underlying the application for the possible transitions.


НАЧАЛО
Оформяне на характеристики и структурата им
Избират се скали за оценяване - ценност, сложност, оценки на екип
Екип оценява характеристиките
Възложител ранжира характеристики
Подреждане по приоритет и отбелязване на зависимости
Формиране на първа реализация (скелет, първа версия)
Оптимизация според приоритетите на проекта
Оформяне на всички реализации
Разпределение на реализациите във времето според възможностите
Необходима дефиниция/смяна на ставки и разходи
Одобрение на плана от възложител
Да
Изготвяне на оферта
Не
Да
Не
Корекция на неодобрен план?
Не, отказ
Приключване на проекта с отказ
Не
Одобрение на оферта
Да
Дефиниране на договор

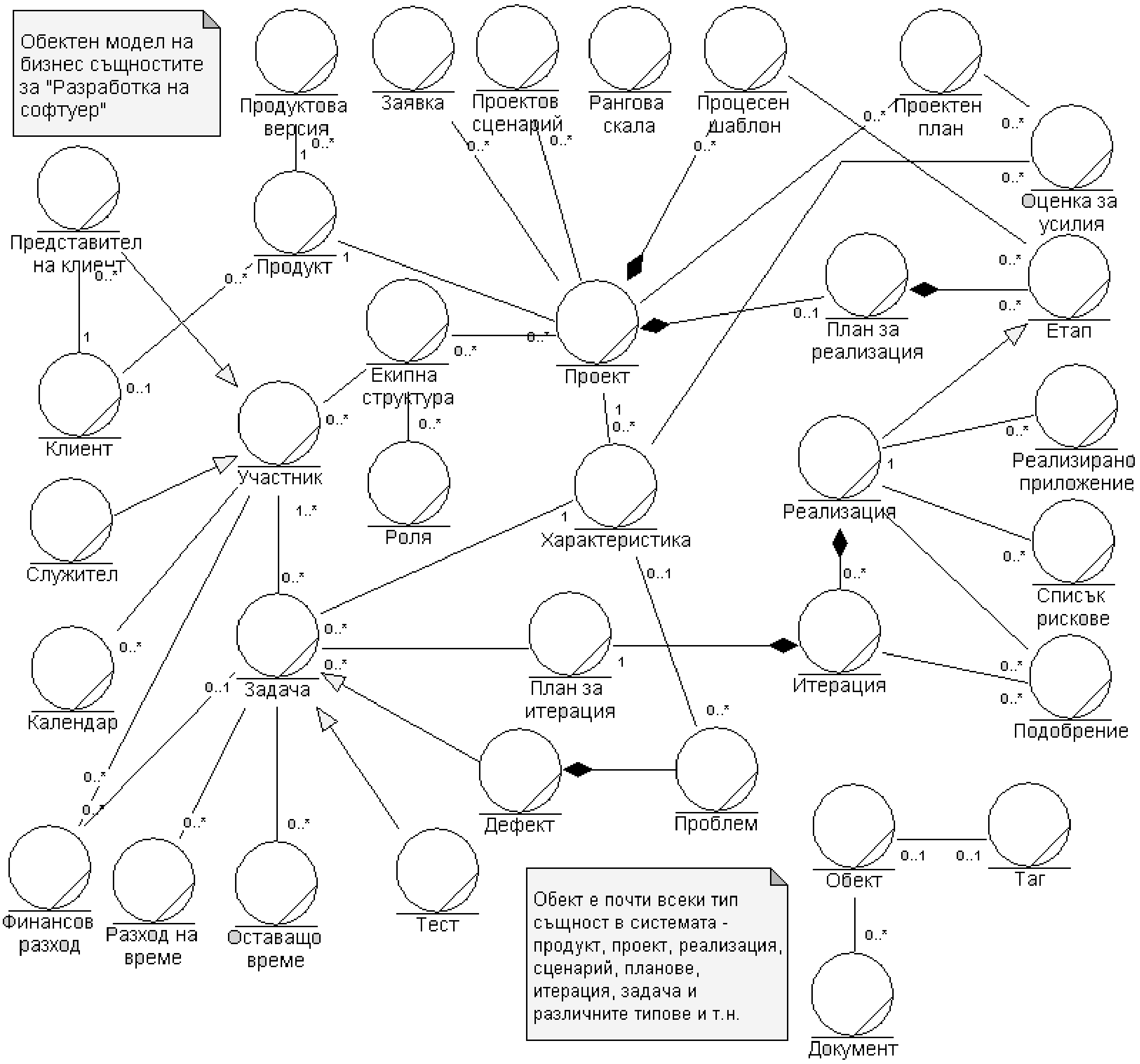
Диаграма на дейностите специфицираща сценария Разработка на продукт\Планиране на проект.

Особености: при изобразяването на дейностите следваме техниката "Бързо планиране" (Blitz Planning) от подход Кристално Ясен, като прескачаме началната дефиниция на характеристики, която е извън този сценарий.

Намирането на първа реализация има за цел да определи продукт, който е или скелет, или е нещо работещо или носи първи приходи.
Оптимизацията за приоритети е според дадените във визията - функционалност, скорост.

Разпределението във времето зависи от капацитет на екипа, ангажименти, реално време на база идеално и скорост, други фактори.

Fig. 3. Activity diagram for BSVS "Product Development"/ "Project Planning" (author's development)


Обектен модел на бизнес същностите за "Разработка на софтуер"

Обект е почти всеки тип същност в системата - продукт, проект, реализация, сценарий, планове, итерация, задача и различните типове и т.н.

Fig. 4. Object model of business entities "Product development" - diagram of business entities and their relationships with other business entities (author's development)

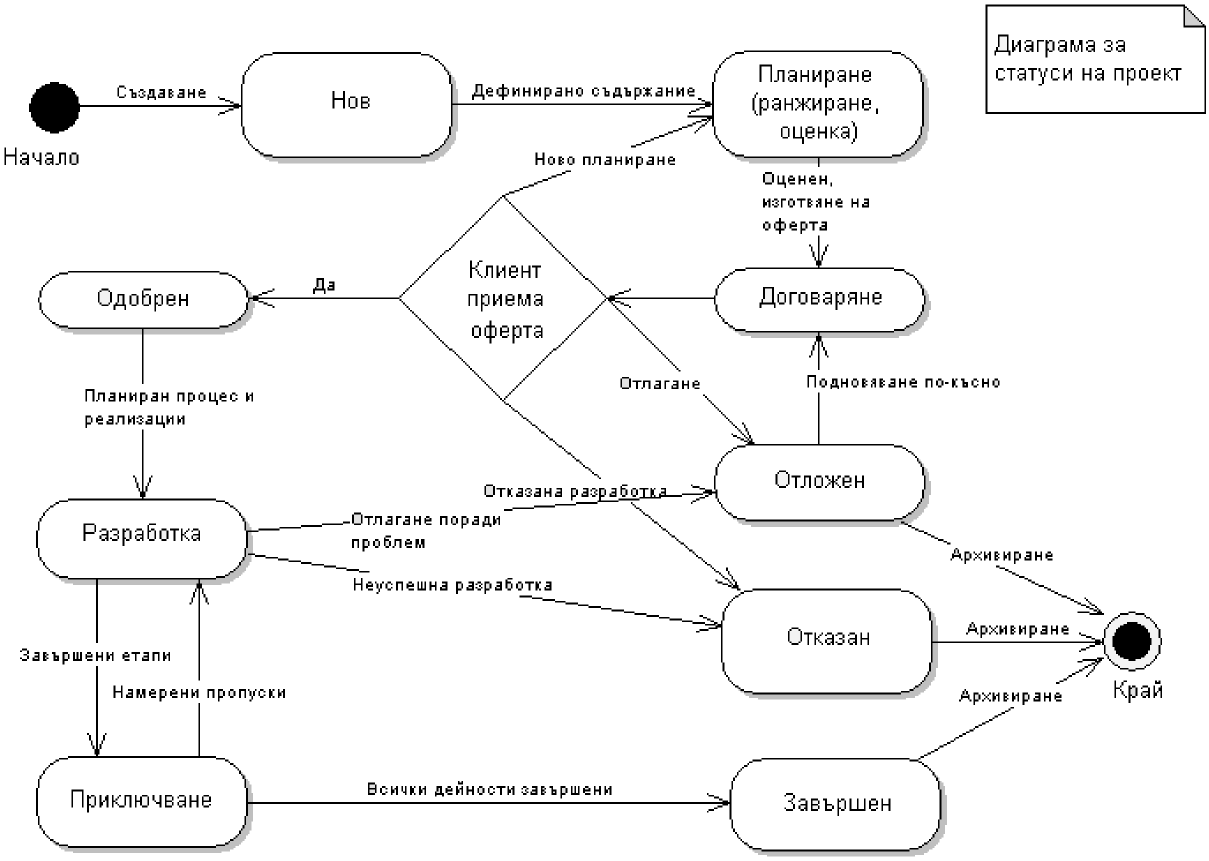
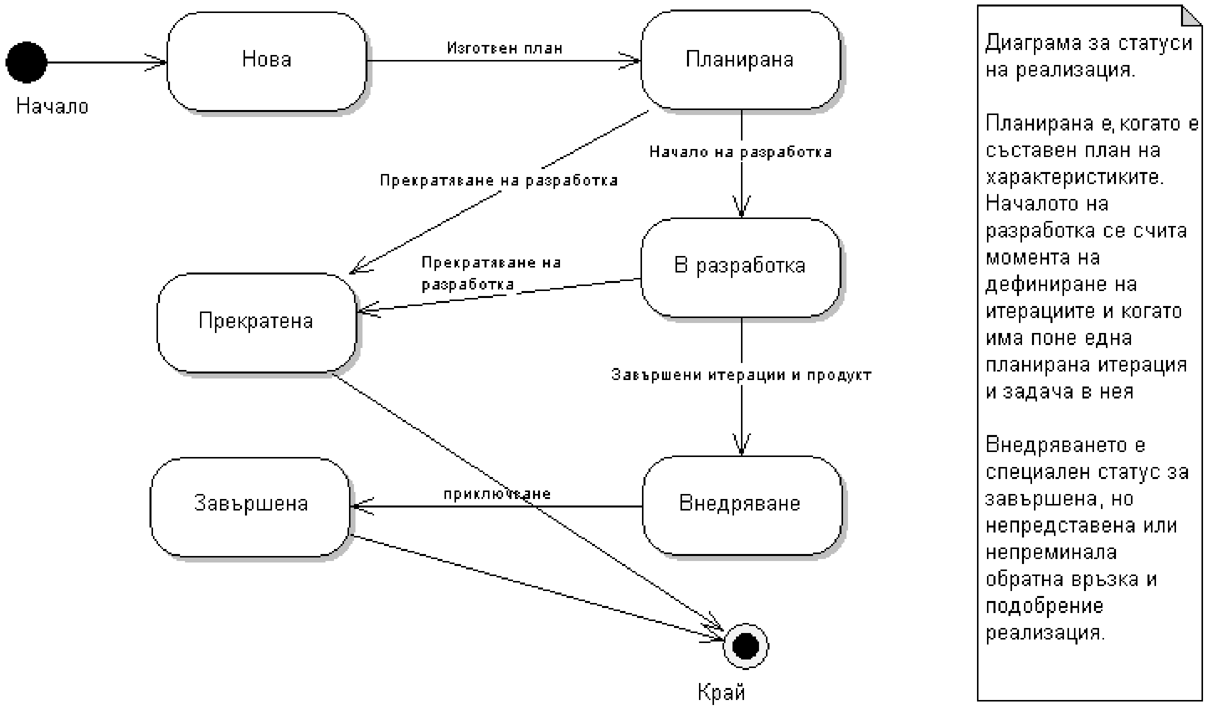


Fig. 5. State diagrams of business entity "Project" (author's development)



Диаграма за статуси на реализация.

Планирана е, когато е съставен план на характеристиките. Началото на разработка се счита момента на дефиниране на итерациите и когато има поне една планирана итерация и задача в нея

Внедряването е специален статус за завършена, но непредставена или непреминала обратна връзка и подобрение реализация.

Fig. 6. Diagrams of the states of the business entity "Realization" (author's development)

Further in the paragraph, the possibilities for improving the information base are considered in terms of: process automation, data transfer and storage, integration with other systems, and the user interface.

In the **second paragraph**, the developed logical model of the system is presented, based on the presented conceptual business model in the concept of the software system.

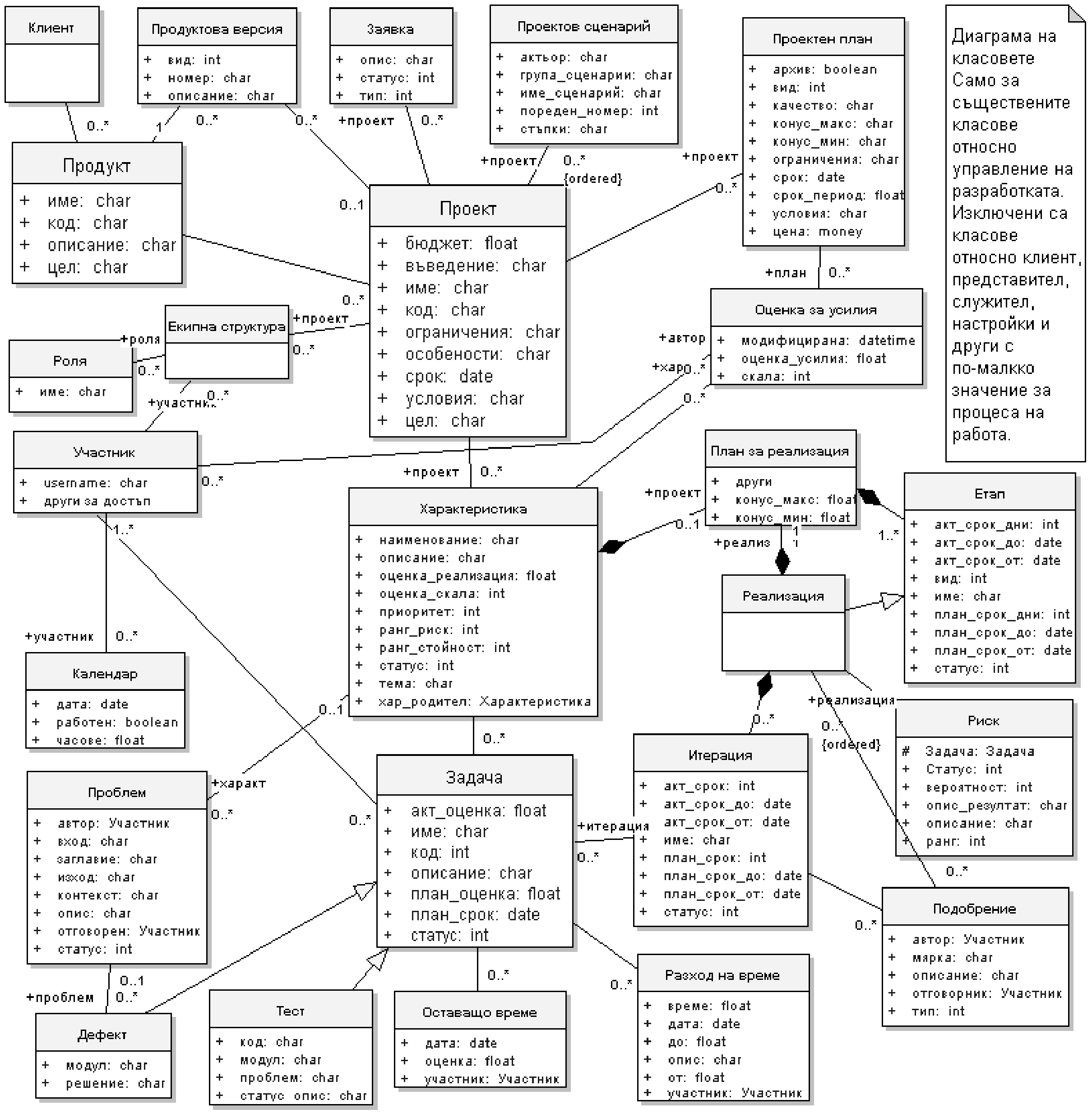


Fig. 7. Class diagram and their relationships for "Product development" (author's development)

A diagram of the main classes implementing the business entities and their relationships are shown in fig. 7 - the essential development classes formed by the business entities, and fig. 8. Under basic we mean those classes that relate directly to development management. Excluded are details on employees, customers, contracts, support, etc. In general, these and similar diagrams are both the basis for developing a database for the software system and a skeleton for the classes in the business logic of the software system.

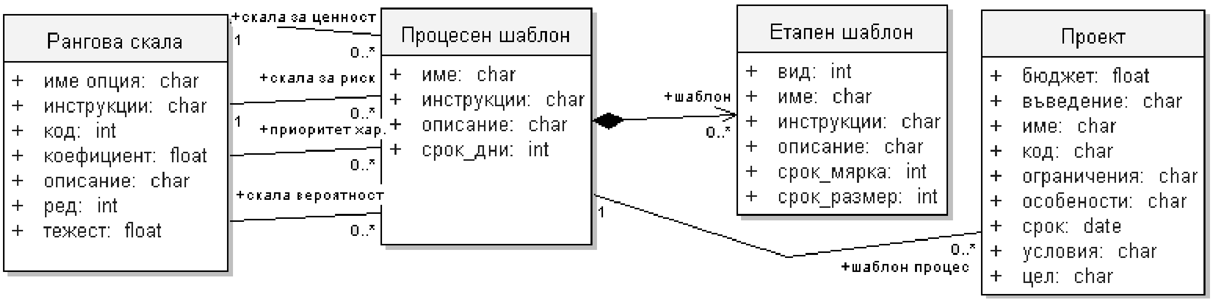


Fig. 8. Essential class diagram for templates formed by business entities (author development)

Regarding the dynamic aspects, diagrams of activities implementing the scenarios were developed, showing the way the process proceeds through activities (fig. 3). These activities are detailed in sequence diagrams and communication diagrams (collaborative, cooperative) for the classes to show how the activities are realized through interaction between the classes over time and their work together. As an example, a sequence diagram is given in fig. 9 with respect to the activity "project planning" based on the activity diagram of fig. 3, i.e. this is a behavior diagram of classes with which the BSVS activity "project planning" is implemented.

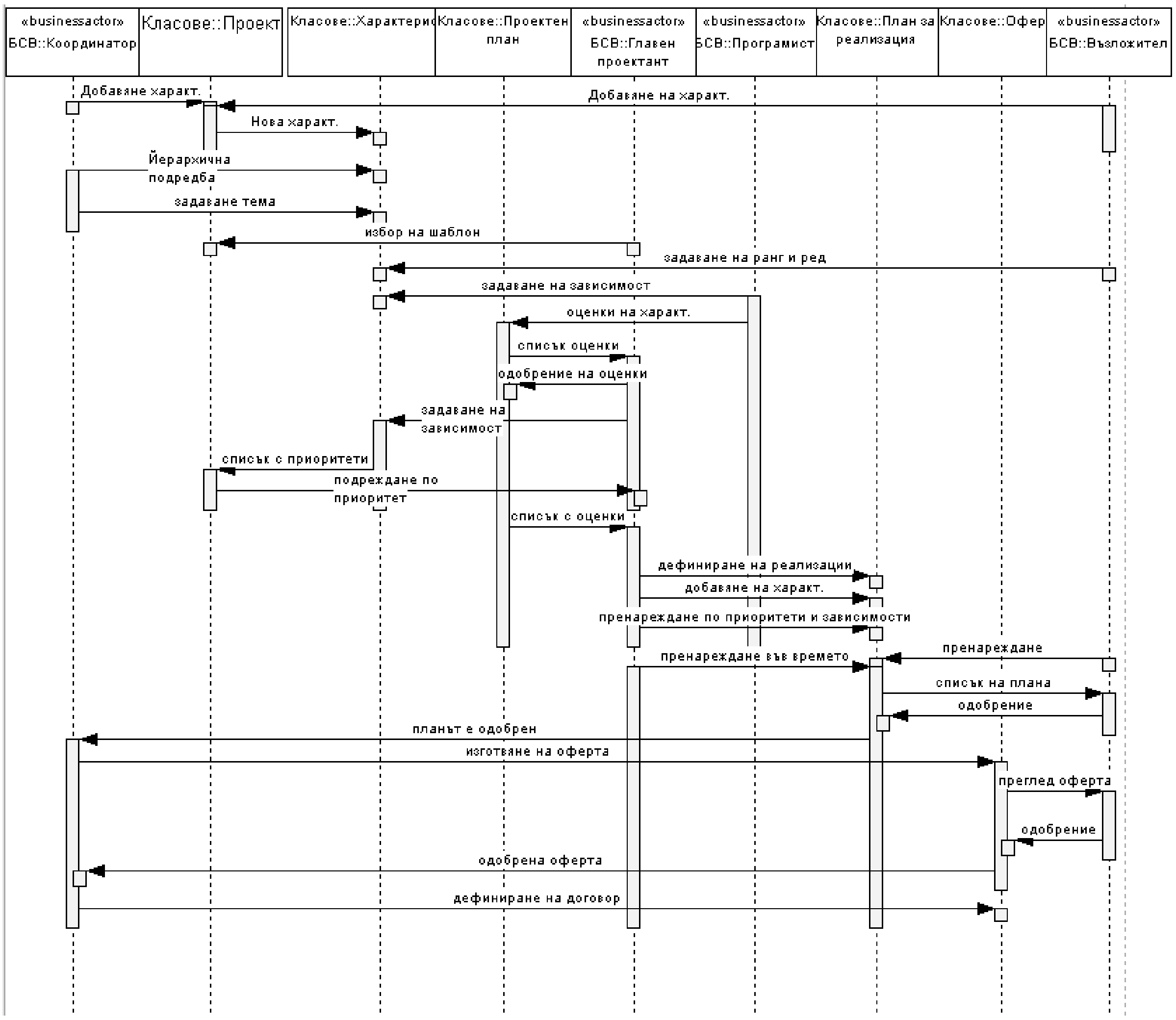


Fig. 9. Interactive diagram for BSVS "Product development/project planning" (author's development)

Another diagram of great importance is given in fig. 10 for sequence of implementation of process template mechanism, for ranking scales for value, risk, probability, degree, etc.[4], to carry out a specific process and arrangement. The figure is implemented by a sequence diagram.

[4] In agile methodologies, quantitative parameters for various indicators are used in the execution of tasks in order to objectify the process of determining work priorities. The rank scales serve this purpose. For example, when compiling the list of tasks (product release backlog) in Scrum, an important place is found in finding the balance between indicators of business value (value) and risk, which is carried out by the owner of the product (Product Owner). Theoretically, these positions are well developed (Wiegers 2005; Wiegers, Beatty 2013). There are a number of examples in the attached plan (eg Lacey 2020).

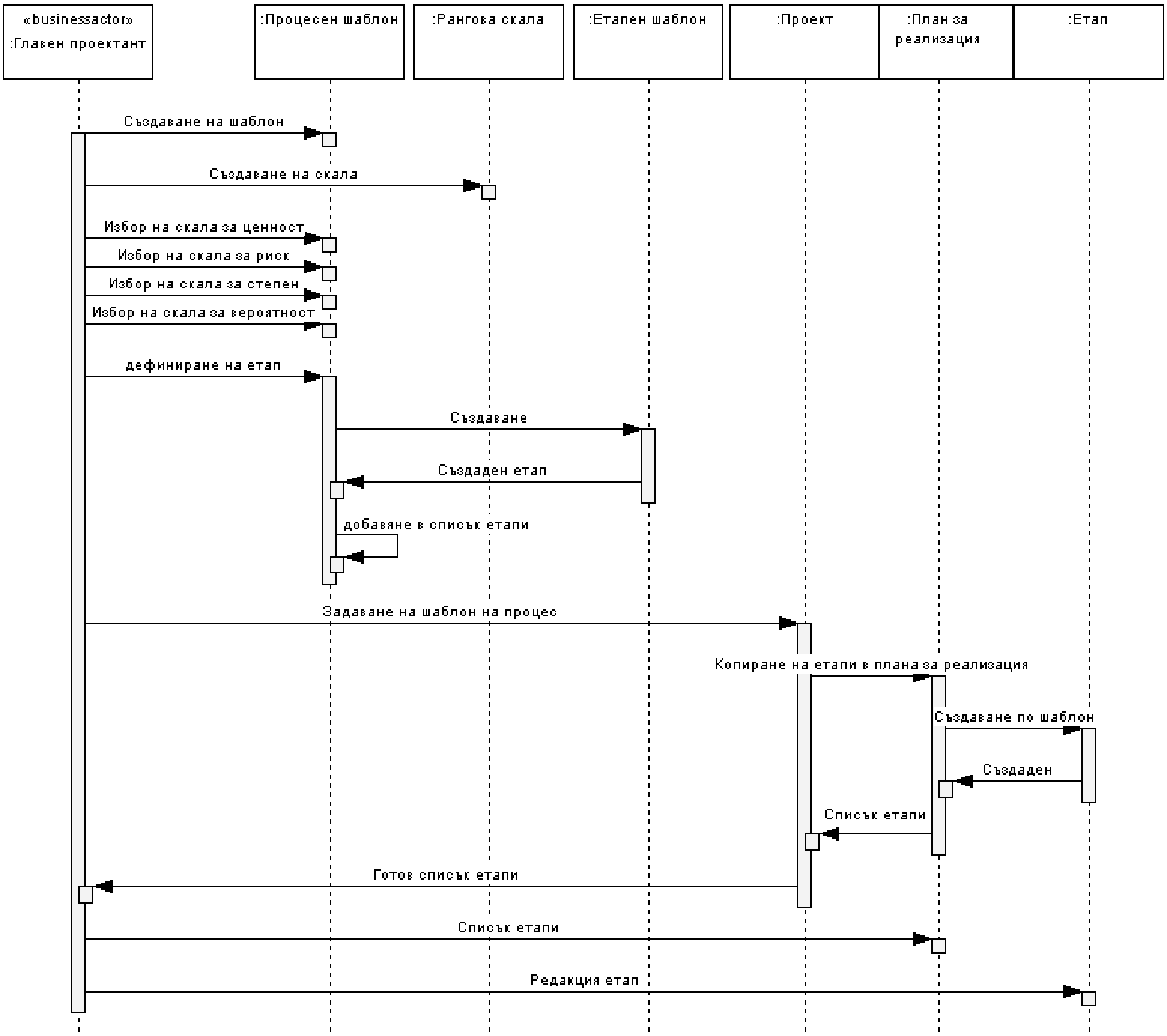


Fig. 10. Diagram for the implementation of process templates (author's development)

In agile methodologies, quantitative parameters for various indicators are used in the execution of tasks in order to objectify the process of determining work priorities. The rank scales serve this purpose. For example, when compiling the list of tasks (product release backlog) in Scrum, an important place is found in finding the balance between indicators of business value (value) and risk, which is carried out by the owner of the product (Product Owner).

Regarding **the system components**, in order to implement the software system according to the selected technology, it is structured into separate logical components. A description of the components, distributed by layers and hardware, is given in table 3.

Table 3

Layout of components by layers and hardware (author's development)

Layer	Hardware/VM	Component	Clarifications
Data layer	Database server	Software system database	To store data about products, projects, processes, users and others from the software system. It is implemented with a relational database
		Document storage	For document storage. It could be a project portal such as MS SharePoint, a configuration management system or a simple file server
	Application server	Data access component	Controls and supports access to data in the database from the business logic
Business logic layer	Application server	A business logic component	Provides logic level functions and rule compliance in the software system. Controls access to data
		A process automation component	It supports automation, implements change and decision rules, and notification rules
		User access component	Users, authorization, assignments, access rights
		Notifications component	To notify users. Generation personally/automatically by the software system, sending in a different form - log, email, etc.
		Reporting component	Generate the data for reports and sections of personal dashboards. Generate reports with settings
Representation layer	Application server	Web server	Site hosting server and web services
		User Job Portal	User access site
		Programmatic Access Component	For programmatic access to web services
		User Interface Components	Front-end components – graphics generation, report rendering, AJAX and other front-end components
	Client machine	A browser to connect to a web application	A browser to connect to the web server and web application of the software system

Our proposed approach requires the creation of **work products (WPs)**, and in a technology startup these may be in the form of documents, but some of them are embedded in the proposed software system.

The **third paragraph** presents Functionality and User Interface. Regarding the components of the representation layer presented in the previous paragraph, the user application interface is defined in detail. There is a common sequence in the screens for individual functions located in the work area. At the top are sections and quick links to other screens in a dedicated page/link bar. Below it is a general information section for a selected product, project, iteration or other, and/or a possible filter for quick transition (if needed). Below them is the review/edit screen itself, with the action buttons usually at the bottom. For all lists, there are buttons/links on the right side for standard quick operations like select, view, refresh, exit, etc.

Regarding the reports, it is appropriate to implement a flexible (easily adaptable) system, allowing both to call up a report from a special report request screen, and directly from a link in another screen with default parameters. In the report request screen, the report group and type are selected, parameters such as product, project, implementation, iteration, employee, term and possibly other private criteria for a given report are set. It is possible to group reports by time - daily, weekly, monthly, quarterly. The report can be requested on screen or its data can be displayed in CSV format. When displayed, it displays a title, selected parameters and grouping, a data table, and a graph section.

Chapter 3. Construction and use of the software development management system in BitPioneers Black Sea Ltd.

In the **first paragraph** of the third chapter, the organization of the activity of the company "BitPioneers Black Sea" Ltd. is presented.

The software system model supporting software development in technology startups is developed in the second chapter. In order to validate the applicability of the detailed architecture and the system proposed in the present study, it should be implemented in a real working environment. At the same time, it is necessary to delineate the boundaries of different approaches to software development in technology start-ups, which are also important in the organization of the activity.

For this purpose, it is necessary to select a company of the type of technology start-up company - application object, which has a suitable subject of activity, has some initial experience in software development and has a need to implement a similar system. The company "BitPioneers Black Sea" Ltd., which the author of the present study has the opportunity to know well in terms of not only the subject of the activity, but also the organization of work, is a suitable object of application. Due to a Non-Disclosure Agreement, as is the usual practice in this highly competitive business, data regarding specific customers, contracts, prices, etc. will not be provided. The exposition treats only professional matters from a scientific point of view.

The company's activity is specialized in the following areas:

1. Digital data collection. The company's team tries, together with its customers, to identify bottlenecks and find simple and effective solutions. Therefore, close cooperation with customers is a key moment for the introduction of such digital solutions. Researching business processes and considering user feedback is at the forefront of developing appropriate software products. Very often, modular software is used, which adapts to the needs of

users.

2. Digitization of work processes. The digitization of work processes enables companies to discover new opportunities for their optimization, as well as to speed up their execution. The digital connection of production capacity and knowledge enables processes to be carried out more operationally, efficiently and more customer-oriented, which becomes a decisive driving force for increasing the competitiveness of medium-sized companies. In the digitalization of work processes, different types of users of the software system are offered a different "point of view" of the process data, so that the users of the system can concentrate on the relevant information at the moment. In this way, the traceability of process steps can be improved and redundant work steps can be avoided or automated, which in turn allows employees to concentrate on important and responsible tasks to increase their efficiency and satisfaction.

3. Remote communication and virtual events. Remote work and communication allow companies to quickly respond to emerging business opportunities and crises, providing operational access to new markets as well as responding to external and internal changes. However, real communication – handshakes and face-to-face communication – has many advantages. BitPioneers enables its clients to transfer these strengths to the digital world by making products tangible and communication personal and uncomplicated by designing customized digital communication solutions. Various software modules are used, combined and adapted to implement such solutions. Capabilities range from scalable TV-quality streams to video conferencing and virtual formats. These can be combined with chats, downloadable content and online voting. Another direction is the creation of virtual 3D worlds that "come alive" with products or services offered by customers, where users can immerse themselves in events, trade fairs or sales halls through augmented or virtual reality and thus connect with the supplier.

As can be seen from the BitPioneers company profile, it is a representative of technology start-up companies involved in software development. We work both on our own projects and on orders, dynamically forming teams that work on specific projects. The outsourcing approach is applied and the so-called "offshore programming". Employees work remotely on the "virtual office" principle. In such situations, in addition to the development and operation of a self-developed software system, specifically designed for the internal needs of the company, issues related to the use of a specific approach in managing software development processes, as well as issues related to the possibility to use virtual tools to complement the underlying software system.

In the **second paragraph**, the plan for realization and implementation of the system developed by the author is presented (Table 4). The set deadlines are indicative and are suitable for small to medium-sized organizations.

Table 4
Project implementation plan for the software production management system in technology start-ups (author's development)

Stage / Implementation	**Idea of the stage**	**Scenarios / features included**	**Sample term**
Detailed design	Project detailing, 360° viewing, conventions	Detailed design stage with project detailing based on current project, concept and chosen approach	One week
Project skeleton	Skeleton of all components and minimum working site access, menu and product test edit	Database with sample data and no optimization. A software project and physical components with only a skeleton of the classes. Server settings, base components added. Interface structure. User login; product overview/definition	Two weeks
Logic and data components	Development of data and business logic components	Product Overview/Definition. Access to data. Enhanced with new modules and module playback	Two weeks
Defining a project and plan	All functions to define a project, content, process and plan; testing the components	A site where a user can define a product, project, features, etc., arrange and evaluate features, select a process and scales for, definition implementations, allocate features, prepare a plan	Three weeks
Work on assignments and testing	Planning and execution of iterations and tasks, testing, issues, defects	Definition of iterations, tasks, list of risks and tasks on them, change of status, minimum plan and time report on them; defines issues, defects, conversion, defect handling and risks	Three weeks
Detailed work report	Plan detailing and reporting on work and iterations	Records spent and remaining time working on tasks. Personal status board. Accounting expenditure of time and money. Notes and documentation to tasks, iterations and all objects. Records findings from reflective improvement in an iteration	Two weeks
Completion of implementation	To be able to complete implementation and reporting. To form reflective improvement	Status and Expected Completion References. Correction of estimates for iterations/realizations, definition of realized products. Records findings from reflective improvement in implementation. Assigns tasks from them. Add requests from client, transfer in features	Two weeks
Complete work cycle and administration	Clearing for a full development cycle. Customer and employee administration.	Improved testing and logging, feature list export for easy documentation. Tags to all objects. Improved time and funds reporting screens. Clients, Representatives, User Administration, Settings, Processes, Scales, etc. Detailed employee data – history, rates, skills, time calendar and more.	Three weeks
Negotiating	To be able to prepare an offer for a client and define a contract;	Add a new contract or offer, define efforts for them, consider capacity. Prepare offer, print, define contract/plan with parameters; detailed time planning by implementations/iterations and capacity compliance	Two weeks

Stage / Implementation	Idea of the stage	Scenarios / features included	Sample term
Accountability		Reporting component, standard performance reports, project completion. Prepares dashboards and other reports	Two weeks
Support	Defining maintenance, monitoring and maintenance work	Definition of maintenance - coordinator/main designer defines requirements, conditions, monitoring during maintenance. Follow up and complete support requests. Convert requests into defects, tasks, or update features. Inventory of deployed versions	Two weeks
Finals	Other extras, closing	A user describes an opinion about satisfaction with the implemented version and the support provided. Bug reports, partition fixed, for maintenance	Two weeks

The plan is for about 26 weeks (or 6 months) and is subject to detailing. The goal is to have a suitable system for operational work on projects in the first three months. When it is ready in three months, the team can test the system with real data.

Further, issues related to the processes of: implementation, operation, administration and coordination are discussed in detail. Regarding the operational management of the team, a set of activities are identified to be carried out: at the beginning of each software project, daily, once or twice a week, at the end of each iteration and implementation, at the end of each project and in emergency situations.

When using a software development management system in a technology start-up company, it is also necessary to choose an appropriate development approach suitable for the specific situation. In our opinion, if an appropriate software system is used, but an inappropriate approach to managing the development processes is applied, high work efficiency will not be achieved, and even a situation of inefficient project work may be reached. For this reason, we believe that the choice of software development approach is as important as the software system used to automate individual software development activities.

In cases of a small company, such as BitPioneers, with a large share of individual ("boutique") developments, in our opinion, it is appropriate to predominantly use the so-called "agile methodologies" (see table 5).

Table 5
Adequate approaches for the individual activities of the BitPioneers company
(author's development)

Type of activity	Approach for small projects	Process, phases, iterations for small projects	Approach for major projects	Life cycle phases and iterations
Client project	CC, SCRUM with short cycles	Light; Initial, Design, 1 - 2 week iterations, Implementation	SCRUM, CC, LSD, DSDM	Initial, Design, 4 -6 week implementations, Implementation
Own products	Unlikely, SCRUM, CC,LSD	Planning, Design, 2 - 4 week implementations, Implementation	SCRUM/UP, CC, LSD, DSDM	Planning, Design, 4 - 6 week implementations, Implementation, Maintenance
Components	The same ones	The same with a modification	SCRUM, CC, LSD	The same with a modification
Support	Modification with regular and occur. tasks	Monthly support with iterations and implementations as needed	Modification with regular and occur. tasks	Monthly support with iterations and implementations as needed

It is possible to choose a combination of two approaches. For example, one may be applied to small client projects, and the second may be entirely for internal products (and components) as well as the largest client projects. The reason is that often successful large projects (less often small ones) continue in some of the following ways:

– maintenance for at least one or two years after completion;

– renewal requests in the next one to two years;

– basis for developing new components – for example for frequent problems;

– a product or vertical reuse system that the BitPioneers company can build as its own based on its experience in projects;

– a base for reusing ready-made solutions in a given area and problem.

In the **third paragraph** of the third chapter, a physical implementation of the system is presented. A selection of technological means was made for the implementation of the system by layers - the data layer, the business logic layer and the representative layer. The possible toolkit for managing the development of the system in several directions is defined - established universal tools for project management, specialized for project management according to a given approach and specialized for software projects/products according to a certain methodology, tied to one group or group of methodologies.

The possibilities of using virtual instruments are considered. Several tables group the virtual tools appropriate for the Preparation, Invention, Business Model, and Software Development phases. Virtual tools provided as services were found to be available for almost every activity. It is a matter of the entrepreneur's judgment which of them and how convenient they are for a given product, idea and business model. In terms of usage cost, most tools have zero or low cost – free versions, free for micro-teams (up to 5 - 10 users) or the price is according to the number of users or consumption volume, which is a big convenience for a startup like BitPioneers because implies low usage costs. A significant problem is the variety of tools to use and the selection of an appropriate tool. In contrast, many of the products have features that cover more than one activity and/or can be integrated with third-party tools for other activities, making it easier to build an integrated startup environment.

In **conclusion**, the dissertation examines issues related to managing the production and maintenance of software products in a technology start-up company. The main place is occupied by the problems of development of the technological start-up company through well-organized production of quality software. The specific activity is described and the problems facing the technology start-up company in relation to key areas - strategy and market positioning, software production and maintenance, resource management (human, financial and information) are outlined.

Small companies are characterized by a lower degree of formalization, simplified procedures and rules. Therefore, regarding the scope of the software system, we assume that software development should cover a certain set of functionalities, avoiding great complexity and getting into purely technical issues regarding design, programming, testing and integration, document management, configurations, etc. n. For them, we suggest using specialized software at individual discretion.

A main quality of the proposed conceptual model of a software system is to have opportunities for many directions of development, and therefore its most important parameters are:

– optimized work for small software projects with single-stage cycles;

– synchronization in the team, automation of decisions and actions to change statuses;

– improved management of time, uncertainty and estimates; automated arrangement of resources, features and tasks over time and based on connections and capacity;

– notifications of problems when the set parameters are expected to be unrealistic;

– variant by type and bilingual support for documentation.

The logical model was developed based on the proposed conceptual business model, and for its implementation, the business entities were initially identified and developed as classes with corresponding properties and relationships.

One of the main points in the development of the software system is the implementation plan, which includes both the planning of the implementations and versions, and the features that are implemented in them, along with the distribution of time. The developed abbreviated version of the work product based on interaction scenarios can be expanded and supplemented, if necessary, according to the needs and strategic views of the management.

When using a software development management system in a technology start-up company, it is also necessary to choose an appropriate development approach suitable for the specific situation, through which to achieve high work efficiency and avoid a situation of project failure. works. Therefore, the process of choosing a software development approach is as important as the software system used to automate the individual software development activities.

The production of software products is a specific activity, and depending on the specific situation, one or the other approach is appropriate. In our opinion, in most cases, the application of an agile software project management approach, based on the CC approach and useful additions from SCRUM, LSD, UP, DSDM, XP, etc., is appropriate for the considered application object. Several options are offered – light for small (client) projects and services, and normal for larger client projects and proprietary products.

The use of virtualization in the operation of the software system can significantly facilitate the start-up software company. With this aim, existing virtual tools and information systems that can be used in the entrepreneurial process, by individual stages and activities, have been studied and described. For each activity, virtual tools are indicated with which it can be performed or assisted. Our research shows that 75% of activities have virtual tools available with medium or high coverage and variety, meaning that virtual tools can handle a large portion of new product development tasks in a software startup. Coverage is lowest in the Invention phase and highest in Software Development.

IV. THE MAIN CONTRIBUTIONS TO THE DISSERTATION

The main contributions of the present study can be summarized as follows:

A. In theory:

- a study was made of the peculiarities of the start-up technology companies developing software, their definition, management and organization of the activity;

– research was done on methodological issues related to organization of activity and management approaches suitable for start-up technology companies.

B. In an attached plan:

– with the use of established formal means, a conceptual and logical model of the software system was developed for technological start-up companies developing software;

– a practical plan for implementing the software system is proposed;

– software technologies are argued to be used depending on the size of the software project implemented by a technology start-up company;

– a summary of the software tools that can be used in the individual stages of the process of creating software in a virtual environment is made.

V. LIST OF THE PUBLICATIONS ON THE TOPIC OF THE DISSERTATION

Articles in Journals

1. **Ivanov, S.**, Petrov, P. (2020). Business scenarios for software system development interaction in a software start-up company. // *Economics and Computer Sciences*, *6*(2), Varna: Knowledge and Business, pp. 27-37.
2. **Ivanov, S.** (2019) Organization of Activities and Management in Starting Software Firms. // *Izvestia Journal of the Union of Scientists - Varna. Economic Sciences Series, 8*(3), Varna: Union of Scientists - Varna, pp.196-207. https://doi.org/10.36997/IJUSV-ESS/2019.8.3.196
3. **Ivanov, S.** (2018) Virtual tools for software development in a software startup. // *Izvestia Journal of the Union of Scientists - Varna. Economic Sciences Series, 7*(3), p.149-160.

Conference Scientific Reports

1. **Ivanov, S.** (2020) Information Systems in Software Startups. Economics, Education and the Real Economy: Developments and Interactions in the Digital Age. // *Collection of reports from the Jubilee International Scientific Conference in honor of the 100th anniversary. from the founding of the IU - Varna: Volume 1*, Varna: Science and Economics, pp. 702-713.
2. **Ivanov, S.**, Petrov, P. (2020) Development of Software Systems by Using Interaction Business Scenarios. // *10th International Conference on Application of Information and Communication Technology and Statistics in Economy and Education (ICAICTSEE – 2020): Conference Proceedings*, Sofia: UNWE, pp.411-419.
3. **Ivanov, S.** (2019) Development Stages of Starting Software Company, Problems and Approaches for Software Development. // *Information and Communication Technologies in Business and Education : Proceedings of the International Conference Dedicated to the 50th Anniversary of the Department of Informatics*, Varna: Science and Economics Publ. House, 2019, 1, pp. 382-396.
4. **Ivanov, S.**, Petrov, P. (2019) Business Models for Starting Software Companies. // *9th International Conference on Application of Information and Communication Technology and Statistics in Economy and Education (ICAICTSEE – 2019): Conference Proceedings*, Sofia: UNWE, pp.170-182.
5. **Ivanov, S.** (2009) Software project management tools with agile methods. // Information technologies in business management. *Collection of reports from an international scientific conference dedicated to the 40th anniversary of the creation of dep. Informatics*, October 16-17, 2009, Varna: Science and Economics, pp. 70-76.