

**Final Project in Requirements Engineering**

**Group 5**

****

**“Tiger”**

**Dorin Bachar- 313603367**

**Marina Shteinfer- 323305458**

**Asaf Schneiderman- 316468636**

**Guy Stein- 308036813**

**srs:**

1. **introduction:**

* **About the organization:** Intel Corporation is an American company that is mostly known to be the largest semiconductor chip manufacturer and one of the developers of the x86 series of instruction sets.
* **Product scope**: The tool's purpose is to reduce working time and increase the team's productivity. As a result, the company's profits will increase.
* **Product value**: The life of the user without the "Tiger" tool was complex. perform tests on the file. He had to type a command containing information about the file and parameters in a precise and specific structure and deal with typing errors. As a result, working hours were extended. The tool will help Intel employees perform tests on the "Layout" files of the chips, by reducing their work times in dealing with typing errors. The tool will identify the necessary information from the open file and the user will only have to select the name of the test and required flags.
* **Intended audience:** Our target audience is a group of Intel employees who perform tests on the layout files of the chips. One of the group members initiated the development of the tool to reduce the work times of the team members. The marketing of the tool will be the presentation of the tool at group meetings.
* **Intended use:** The user will open the file on which he wants to perform the tests. After that, it will open the "Tiger" tool which identifies the necessary information from the file, for example, the name of the library, and the name of the file. The user will have to choose the name of the test from a built-in database (Combobox) and the flags. Then he will press the "Run" button which will build the command to run and execute it. Finally, a test results report will be presented.

If the user wants to stop a test in the middle, he can press the "stop flow" button.

* **Definitions and acronyms:**

1. “**Tiger”** is the tool’s name that the users of the organization will use to perform the tests (DRC, LVS) they are familiar with on the files.
2. **The layout file** is a design file of the chip described by metal layers, transistors, diodes, etc.

* **Table of contents:**

**System requirements and functional requirements ……………4**

**Non-functional requirements………………………………………. 4**

**Alternatives………………………………………………………………4**

**The essence of the organization………………………………………5**

Interview Summary……………………………………………………5

The Main Persona…………………………………………………..…5

Prototype………………………………………………………………5

Mov……………………………………………………………………..6

**Organization requirements and initial architecture……………………….7**

The initial architectural model…………………………………………..7

SOA principles……………………………………………………………7

Design patterns………………………………………………………….7

Risks table……………………………………………………………….8

Risks table for Technical debt………………………………………...9

Testing table………………………………………………………….10

**Cognition and motivation theories…………………………………….11**

Distributed cognition……………………………………………….11

Diagram…………………………………………………………….11

Flow characteristics……………………………………………….12

SDT (self-determination theory)...............................................12

Gamification………………………………………………………13

**Reference to feedback……………………………………………….13**

**Challenges during the working process……………………………14**

**Reference……………………………………………………………………..14**

**2. System and functional requirements:**

* The tool can run a test command.
* The tool can insert parameters for the test.
* The tool can click the buttons in the GUI.
* The tool can stop the test in the middle.
* The tool creates a test command.
* The tool creates a test results report.
* The tool displays a message at the end of the test.
* The tool displays a report of the number of uses of the system.

**3. Non-functional requirements (quality attributes):**

* The parameters are selected from a built-in database. (Backup)
* The parameters are the name of the test to be performed, and the selection of "flags" suitable for the desired test. (usability)
* The buttons in the GUI are: "Run" which creates the command line and runs the test, and "stop flow" which stops the test in the middle. (Operability)
* The test results report shows if there are errors in the test for the scheme or if everything went successfully. (Reporting)
* Ending message of the tool including the name of the test, and the name of the layout file on which the test was performed. (Data integrity)
* The tool is adapted to different work environments. (Adaptability)

**Alternatives:**

As part of the project, we chose 3 potential alternatives:

1. **Microsoft Azure cloud alternative** - Azure DevOps service (PaaS).

This alternative enables the purchase of a platform for writing the tool's code.

That is, it gives us servers and software ready to write the code.

There are advantages to saving money on server maintenance, storage, and support, but the downside is that the organization's developers will still have to support the tool.

2. **Cadence software** -Purchase ready-made software. This is the cheapest option, but for this reason, it is less flexible and reliable because it is difficult to adapt it to the work environment of the organization.

3. **The software house "My World" company** - A software house for the development of the tool. This is the most expensive option.

On the one hand, you need to hire the services of an external team for development, support, and adding features.

On the other hand, the organization gives up recruiting and training employees for the development of the tool.

**4. The essence of the organization:**

**a. Interview Summary:**

We interviewed two people from the Intel Company (stakeholders). The first one is the user and initiator of the tool and the second one is the programmer of the tool.

The conclusions are:

* The tool is very useful, effective, and contributes to the way of working.
* The users use the “Tiger” daily.
* They want to continue to develop the tool.
* There are complaints that the tool doesn’t work in all environments.

**b. The Main Persona:**

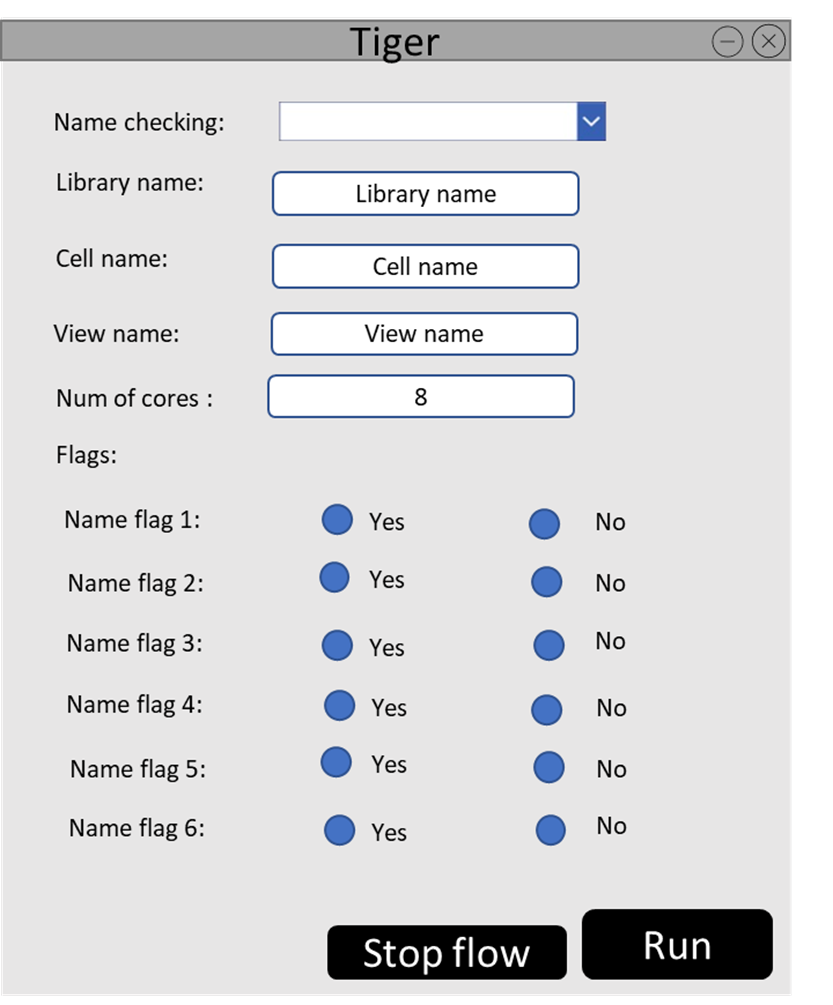
Adi, 35.

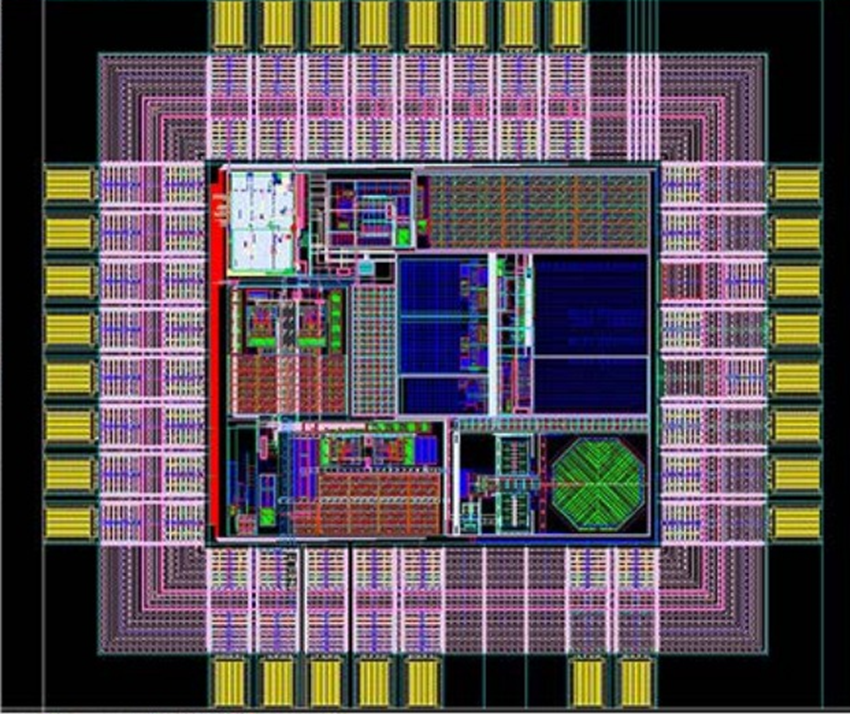
**Background:** Practical Electronic Engineer works for Intel as project

manager.

**Description:** Adi is interested in gadgets and new technology that is constantly improving and advancing. In addition, he admires the characters in "Marvel" films. He finds problems and incorrect ways of working and initiates ideas to improve his and his team members' working way to save time. He initiated the development of the "Tiger" tool.

**c. Prototype:**





**d. MOV**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FR** | **NFR** | **Area** | **Matric** | **Timeframe** | **Statement** |
| The tool creates a testing command. | Pressing the “Run” bottom creates the command without errors. | Time  Financial | Shortening work time  Prevention human errors while typing the command | Developing time will be 6 weeks (3 sprints)  After a month there will be an improvement in users' performance and projects will be completed in time or before the deadline. | The project will be successful if the tool will reduce the time of typing a command by 80% within a month. |
| The tool allows users to display a report of the number of uses in “Tiger”. |  | Organizational | The number of users in the tool.  The number of login to the tool. | In a month there will be more users using the tool. | The project will be successful if the number of users in the tool will be more than 30 per day within a month. |

**5. Organization requirements and initial architecture:**

**a. The initial architectural model:**

Event-driven architecture

Since the user enters details for the test to be performed and presses the "RUN" button when he wants to perform the test or the “stop flow” button when he wants to stop the test, therefore it is important to respond to events when entering the parameters and clicking.

**b. SOA principles:**

|  |  |
| --- | --- |
| SOA principles | Description |
| Service abstraction | Service hides the logic from the outside world. It is often dropped as a principle.  The user doesn’t have access to the server. |
| Service Composability | Service breaks the big problem into smaller problems. Each server runs a different test. |
| Service Interoperability | Services should use standards that allow diverse subscribers to use the service.  Our tool is used by users from all over the world. |
| Discoverability | Service can be discovered. In our tool, the parameters for the test can be searched in the build database. |

**c. Design patterns:**

**singleton** - running the same checking command multiple times by several users.

**adapter** - different information will be displayed in the results report when the “layout” file isn’t the same.

**d. Risks table:**

|  |  |  |  |
| --- | --- | --- | --- |
| The risk | Type | Probability | Effect |
| The staff will refuse to work with the tool and they will want to work the old way. | People | Medium | tolerable |
| Departure of a programmer during development. | People | Low | tolerable |
| Many will report problems with the tool. | Technology | Low | catastrophic |
| There will be no adaptation  of the tool to the work environment. | Technology | Medium | Serious |
| Users will ask for another tool that will be faster. | Tools | Low | tolerable |
| Management will regret developing the tool in the middle of development. | Organizational | Low | catastrophic |
| The tool will not be ready with a deviation of 30% or more from the planned. | Estimation | Low | Serious |
| Failure to meet troubleshooting deadlines. | Estimation | Medium | Serious |
| Requirements of the tool may change during development. | Requirements | Medium | tolerable |

**e. Risks table for Technical debt:**

|  |  |  |  |
| --- | --- | --- | --- |
| **num** | **Technical debt** | **Solution 1** | **Solution 2** |
| **1** | Pressure from the users to release the tool so that they can work with it and promote projects, but in practice there may be a situation where we get a tool that does not work in all work environments. | Release the appropriate tool for a work environment where most of the group members work and then later update to the new environment. |  |
| **2** | Lack of conformity to the standards that the users expected  For example: not all the names of the tests appear. | Users will be able to perform partial checks on the file. | Updating the tool for additional tests (existing and new). |
| **3** | Lack of a test suite for the tool that can prevent bugs. | When building the tool try to cover as many edge cases as possible. |  |
| **4** | Lack of understanding of the tool's development needs and the difficulties during its development, for example: developing the tool in several programming languages, which causes the tool to slow down | Dealing with user support and trying to improve the tool by making a transition of writing the tool in a uniform language or optimizing the code. | Provide more processors so that they can run the test and then it will run faster. |
| **5** | A lack of code documentation creates a problem when the developer of the tool is no longer present in the company and then the code must be understood, which takes time and difficulty. | Going over the code and documenting the code in the critical sections that are usually modified suchas: adding the name of the test and flags. |  |

**f. Testing table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Type** | **Manual/Automated** | **Expected result** |
| **1** | Entering invalid parameters such as the file in view name isn’t of layout type. | system | automated | Displayed an error message to the user:  “The file in view name isn’t of layout type ”. |
| **2** | Entering invalid parameters such as the number of cores is incorrect,i.e too high or too low. | unit | automated | Displayed an error message to the user:  “The number of cores is incorrect ”. |
| **3** | opening of the tool without the “layout” file open. | system | automated | Open error window:  “please, open  “layout” file”. |
| **4** | Insert an error in the “layout” file. | system  +acceptance | automated | Expect to see the error in the results report. |
| **5** | pressing a “Run” button. | unit +acceptance | automated | A log file will be opened which shows a report of the results of the checking at the end. |
| **6** | testing the load of the tool by using the tool with several users at the same time. | system | automated | The tool will stand loaded and work as usual. |
| **7** | pressing a “stop flow” button. | unit | automated | A message window opens: “The test stopped in the middle” |

**6. Cognition and motivation theories:**

**a. Distributed cognition:**

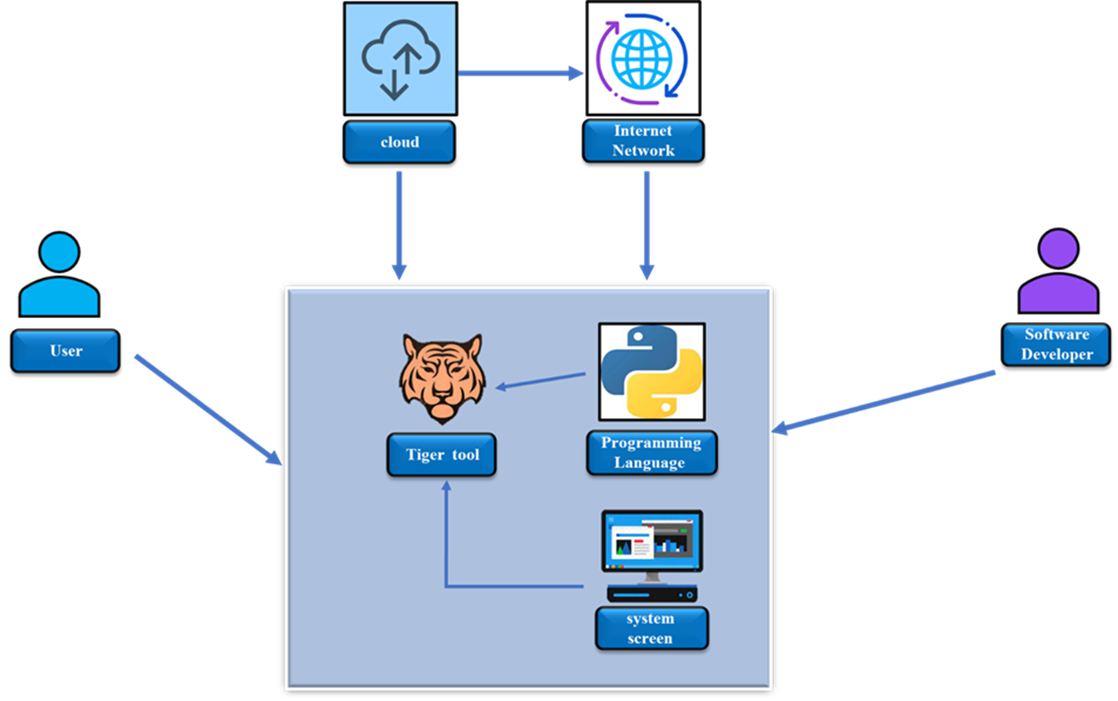
**Memory distributed among units-** The parameters are selected by the user from a built-in database, so there is no need to use memory.

**External/internal representations-** graphical representation of the tool.

**Each unit has its own cognition-** The unit that works with the tool in the organization consists of different levels of employees: engineers, and particle engineers.

Adjusting and formulating the system requirements for all types of users in a way that everyone can understand its use.

**Culture affects the actions of the units-**The organization and encourages its employees to constantly improve and innovate technologies, therefore the process of assimilating the tool to system users works more smoothly.



**b. Flow characteristics:**

* **Concentration -** The tool is in one window so the user's concentration is focused on one screen and he doesn't need to switch windows.
* **Challenge -** Choosing the name of the test to run and dealing with the result report.
* **User skills -**  The users of the tool should be familiar with the tests and their purpose, for example, the Layout designer.
* **Control -** The user can choose the test to run and the file on which the test is performed.
* **Clear goals -**The clear goal is to run the test on the file and get a results report.
* **Immersion-** Following the results of the questionnaires filled out by the users, the process of assimilating the tool into the system was successfully carried out for the users.
* **social interaction-** Other users can check someone else’s files and help them to understand the error shown in the result report.

**c. SDT (self-determination theory):**

* **Competence -** Users feel competent when the results report is without errors or when there are errors they know how to deal with.
* **Autonomy -** The user can choose the test to run and the file on which the test is performed.
* **Relatedness -** other users can check someone else’s files and help them to understand the error shown in the result report. In addition, the desire to finish the test in a faster way and meet the team’s deadline.

**d. Gamification:**

* **Scoring -** displaying a scoring table for the top users who have the fewest errors when running the tests. The table will cause recognition and appreciation for the employees and will stimulate competitiveness among the employees.
* **Badges -** receiving badges based on use with the tool, the more you use the tool, the more badges will increase. Designed and advanced badges will indicate that employees are more experienced in working with the tool, which may motivate them to make many uses of the tool and help other employees in working with the tool because they are already labeled as "experts" in working with the tool in the system.
* **Survey/Rating -** Receiving the opinion from the users about the tool by conducting a survey or rating of the tool, which is a pop-up window requesting to make a rating/opinion for the use of the tool. However, this abrasive may be distressing for the user as it may impair the continuity of work with the tool in the system.

**7. Reference to feedback:**

We made changes according to the feedback that we got:

* We added a time frame to the statements in our MOV.
* We added the “Discoverability” principle to our SOA.
* We tried to make the project more specific and shorter.

**8. Challenges during the working process:**

* **Synchronize our schedules** - All the group members had different schedules for example some of them had a job and some of them had more courses. As a result, we had a difficult time finding time to meet everyone. To solve this problem some time we worked individually or met in smaller groups.
* **Disagreements between group members** - Because we are all different people with different opinions, sometimes we didn’t think the same about some of the ideas we talked about in the project. To solve these disagreements, we debated a lot and made democratic decisions.
* **Lack of time** - Summer courses are shorter than regular semesters so we didn’t have much time from the start. To make the most of the time we had, we tried to meet as much as possible, even if it was just part of the group, and divided the work between us.
* **To get users to answer the questionnaires** - It was hard to get the Intel workers to answer the questionnaires because they were also very busy and tired after work. We only could remind them of every opportunity we got.

**9. Reference:**

<https://www.ic-cracker.com/integrated-circuit-layout-design-function/>

<https://forms.gle/zLmChgwCPZnr6yHr8>

[https://en.wikipedia.org/wiki/Intel](https://forms.gle/zLmChgwCPZnr6yHr8)

[https://www.methoda.cloud/?section=254#section-24](https://forms.gle/zLmChgwCPZnr6yHr8)

[https://migdala.com/%D7%97%D7%A9%D7%99%D7%91%D7%94-%D7%A2%D7%99%D7%A6%D7%95%D7%91%D7%99%D7%AA-%D7%90%D7%99%D7%9A-%D7%96%D7%94-%D7%A2%D7%95%D7%91%D7%93/](https://forms.gle/zLmChgwCPZnr6yHr8)