

## ДОДАТОК А

## ЛІСТИНГ КОДУ

```

"use strict";
const _ = require('lodash');
const uuid = require('uuid/v4');
const moment = require('moment');

const prepareGraphAndOrder = (tasks) => {
  const idsArray = [];
  tasks.forEach(task => {
    if(!task.predecessor.length){
      idsArray.push(task.taskId);
    }
  });

  tasks = tasks.map(task => _.omit(task, ['orderId']));

  const graph = [];
  const tasksAmount = tasks.length;
  for(let i = 0; i < tasksAmount; i++){
    graph.push(Array(tasksAmount).fill(0, 0, tasksAmount))
  }

  let pointIndex = 0;
  let idsArrayIndex = 0;
  while(true){
    const currentId = idsArray[idsArrayIndex];
    const taskIndex = tasks.findIndex(task => task.taskId === currentId);
    if(!tasks[taskIndex].orderId){
      tasks[taskIndex].orderId = pointIndex;
      pointIndex++;
    }

    tasks.forEach(task => {
      if(task.predecessor.includes(currentId)){
        if(!task.orderId) {
          task.orderId = pointIndex;
          pointIndex++;
        }
      }
      graph[tasks[taskIndex].orderId][task.orderId] = 1;

      if(!idsArray.includes(task.taskId)) {
        idsArray.push(task.taskId);
      }
    }
  });

  idsArrayIndex++;
  if(idsArrayIndex === tasksAmount){
    break;
  }
}

```

```

tasks.sort((task1, task2) => task1.orderId > task2.orderId);
return {
  tasks,
  graph
};
};

const countDirect = (tasks, graph) => {
  for( let i = 0; i < tasks.length; i++){
    if(i === 0){
      tasks[i].earlyStart = 0;
      tasks[i].earlyFinish = tasks[i].estimateMax;
    } else {
      let maxValue = 0;
      for(let j = 0; j < i; j++){
        if(graph[j][i] && tasks[j].earlyFinish > maxValue){
          maxValue = tasks[j].earlyFinish;
        }
      }
      tasks[i].earlyStart = maxValue;
      tasks[i].earlyFinish = tasks[i].earlyStart + tasks[i].estimateMax;
    }
  }
  return tasks;
};

const countProjectDuration = (tasks) => {
  return Math.max(...tasks.map(task => task.earlyFinish));
};

const countBackward = (tasks, graph) => {
  for( let i = tasks.length - 1; i >= 0; i--){
    const isAtTheEnd = graph[i].reduce((a, b) => a + b) === 0;
    if(isAtTheEnd){
      tasks[i].lateFinish = tasks[i].earlyFinish;
      tasks[i].lateStart = tasks[i].lateFinish - tasks[i].estimateMax;
    } else {
      let minValue = Number.MAX_SAFE_INTEGER;
      for (let j = i; j < tasks.length; j++){
        if(graph[i][j] && tasks[j].lateStart < minValue){
          minValue = tasks[j].lateStart;
        }
      }
      tasks[i].lateFinish = minValue;
      tasks[i].lateStart = tasks[i].lateFinish - tasks[i].estimateMax;
    }
  }

  return tasks;
};

const countCriticalPath = (tasks) => {
  for (let i = 0; i < tasks.length; i++){
    tasks[i].isOnCritPath = tasks[i].lateFinish === tasks[i].earlyFinish;
  }
  return tasks;
};

```

```

};

const countFloats = (tasks) => {
  for (let i = 0; i < tasks.length; i++){
    tasks[i].startFloat = tasks[i].lateStart - tasks[i].earlyStart;
    tasks[i].finishFloat = tasks[i].lateFinish - tasks[i].earlyFinish;
  }

  return tasks;
};

const countFreeFloat = (tasks, graph) => {
  for(let i = 0; i < tasks.length; i++){
    let minSuccessorEarlyStart = Number.MAX_SAFE_INTEGER;
    for(let j = i; j < tasks.length; j++){
      if(graph[i][j] && tasks[j].earlyStart < minSuccessorEarlyStart){
        minSuccessorEarlyStart = tasks[j].earlyStart;
      }
    }
    tasks[i].freeFloat = minSuccessorEarlyStart - tasks[i].earlyStart -
tasks[i].estimateMax;
  }
  return tasks;
};

function prepareUpdateStringAndObject(inputObject){
  let updateString = 'set ';

  const expressionAttributeValues = {};
  const expressionAttributeNames = {};
  Object.keys(inputObject).forEach(key => {
    updateString += `#${key}= :${key},`;
    expressionAttributeValues[`${key}`] = inputObject[key];
    expressionAttributeNames[`${key}`] = key;
  });

  updateString = updateString.slice(0, -1);

  return {
    updateString,
    expressionAttributeValues,
    expressionAttributeNames
  }
}

const getParallelIndexes = (index, points, graph) => {
  const parallelIndexes = [];
  let prev, next;
  //searching prev
  for(let i = 0; i < graph.length; i++){
    if(graph[i][index]){
      prev = i;
    }
  }
  //searching next
  for(let i = 0; i < graph.length; i++){
    if(graph[index][i]){

```

```

    next = i;
  }
}
for(let i = 0; i < graph.length; i++){
  if(prev &&
    next &&
    graph[i][next] &&
    graph[prev][i] &&
    i !== index &&
    points[i].isOnCritPath){
    parallelIndexes.push(i);
  }
}
return parallelIndexes;
};

const getParallelCrashAvailable = (indexes, actions) => {
  if(indexes.length) {
    let isAvailable = true;
    indexes.forEach(index => {
      if (!(actions[index].estimateMax - actions[index].estimateMin)) {
        isAvailable = false;
      }
    });
    return isAvailable;
  }
  return false;
};

const countCrash = (tasks, graph, overheadCostPerDay) => {
  let runNumber = 0;
  const result = [];

  const activityDirect = countDirect(tasks, graph);

  const projectDuration = countProjectDuration(activityDirect);

  const activityBackward = countBackward(activityDirect, graph);

  const activityFloats = countFloats(activityBackward);

  let activityCritPath = countCriticalPath(activityFloats);

  const daysCost = overheadCostPerDay * projectDuration;

  let activityCost = 0;

  for(let i = 0; i < activityCritPath.length; i++){
    activityCost += activityCritPath[i].taskCost;
  }

  const projectCost = daysCost + activityCost;

  //setting initial
  result[0] = {};

  for (let i = 0; i < activityCritPath.length; i++){

```

```

    result[0][activityCritPath[i].taskId] = {
      estimateMax : activityCritPath[i].estimateMax,
      estimateMin : activityCritPath[i].estimateMin,
      crashCost : activityCritPath[i].crashCost,
      TF : activityCritPath[i].finishFloat
    };
  }

  result[0].crashId = 'initial';
  result[0].cost = null;
  result[0].save = null;
  result[0].projectCost = projectCost;
  result[0].projectDuration = projectDuration;
  result[0].runNumber = runNumber;
  runNumber++;

  while(true){
    const runResult = {};
    for(let i = 0; i < activityCritPath.length; i++){
      runResult[activityCritPath[i].taskId] = {};
    }
    //checking if crash is available
    let availableCrashDuration = 0;
    for(let i = 0; i < activityCritPath.length; i++){
      if(activityCritPath[i].isOnCritPath) {
        availableCrashDuration += activityCritPath[i].estimateMax -
activityCritPath[i].estimateMin;
      }
    }
    if(!availableCrashDuration){
      break;
    }
    //searching for cheapest crash

    let cheapestCrashIndex;
    let hasParallel = false;
    let cheapestCrashCost = Number.MAX_SAFE_INTEGER;
    for(let i = 0; i < activityCritPath.length; i++){
      if(activityCritPath[i].isOnCritPath &&
        activityCritPath[i].estimateMax - activityCritPath[i].estimateMin
> 0 &&
        activityCritPath[i].crashCost < cheapestCrashCost){
        cheapestCrashIndex = i;
        cheapestCrashCost = activityCritPath[i].crashCost;
      }
    }

    const parallelIndexes = getParallelIndexes(cheapestCrashIndex,
activityCritPath, graph);
    const isParallelCrashAvailable =
getParallelCrashAvailable(parallelIndexes, activityCritPath);
    if(parallelIndexes.length){
      if(isParallelCrashAvailable) {
        hasParallel = true;
      } else {
        break;
      }
    }
  }

```

```

}

const updatedActions = [];
for(let i = 0; i < activityCritPath.length; i++){
  if(i === cheapestCrashIndex || parallelIndexes.includes(i)){
    runResult[activityCritPath[i].taskId].daysSaved = 1;
    runResult[activityCritPath[i].taskId].estimateMax =
activityCritPath[i].estimateMax - 1;
    updatedActions.push({
      estimateMax : activityCritPath[i].estimateMax - 1,
      taskId : activityCritPath[i].taskId,
      estimateMin : activityCritPath[i].estimateMin,
      crashCost : activityCritPath[i].crashCost
    });
  } else {
    runResult[activityCritPath[i].taskId].daysSaved = 0;
    runResult[activityCritPath[i].taskId].estimateMax =
activityCritPath[i].estimateMax;
    updatedActions.push({
      estimateMax : activityCritPath[i].estimateMax,
      taskId : activityCritPath[i].taskId,
      estimateMin : activityCritPath[i].estimateMin,
      crashCost : activityCritPath[i].crashCost
    });
  }
}

const updatedActivityDirect = countDirect(updatedActions, graph);
const updatedProjectDuration =
countProjectDuration(updatedActivityDirect);
const updatedActivityBackward = countBackward(updatedActivityDirect,
graph);

const updatedActivityFloats = countFloats(updatedActivityBackward);
const updatedActivityCritPath =
countCriticalPath(updatedActivityFloats);

for(let i = 0; i < updatedActivityCritPath.length; i++){
  runResult[updatedActivityCritPath[i].taskId].TF =
updatedActivityCritPath[i].finishFloat;
}

if(!hasParallel) {
  runResult.cost = cheapestCrashCost;
} else {
  runResult.cost = 0;
  for(let i = 0; i < updatedActivityCritPath.length; i++){
    if (i === cheapestCrashIndex || parallelIndexes.includes(i)){
      runResult.cost += updatedActivityCritPath[i].crashCost;
    }
  }
}

runResult.save = overheadCostPerDay;
runResult.projectCost = result[result.length - 1].projectCost +
runResult.cost - overheadCostPerDay;
runResult.projectDuration = updatedProjectDuration;
runResult.crashId = uuid();

```

```

    runResult.runNumber = runNumber;
    runNumber++;
    result.push(runResult);
    activityCritPath = updatedActivityCritPath;
  }

  return result;
};

function checkAvailability(projectParams, key, worker){
  const vacationsDiff = worker.vacations.map(vacation =>
moment(projectParams[key].teamStartDate).diff(moment(vacation.endDate)));
  let isFree = true;
  vacationsDiff.forEach(diff => {
    if(diff < 0) {
      isFree = false;
    }
  });

  if(worker.project){

if(moment(projectParams[key].teamStartDate).diff(moment(worker.project.en
dDate)) < 0){
  isFree = false;
}
}
  return isFree;
}

function prepareTeam(projectParams, workers){
  const result = {};

  Object.keys(projectParams).forEach(key => {
    result[key] = [];
    const ratings = {};

    const appropriateDevelopers = workers.filter(worker => {
      const isMainSkillSame = worker.mainSkill ===
projectParams[key].mainSkill;
      const isSeniorityAppropriate =
Object.keys(projectParams[key].seniority).includes(worker.seniorityLevel);
      const isAvailableOnStartDate = checkAvailability(projectParams,
key, worker);

      let isOk = true;
      [isMainSkillSame, isSeniorityAppropriate,
isAvailableOnStartDate].forEach(condition => {
        if(!condition){
          isOk = false;
        }
      });
      return isOk;
    });

    appropriateDevelopers.forEach(developer => {

```

```

    const skillRating = _.intersection(projectParams[key].skills,
developer.skills).length / projectParams[key].skills.length;

    if(ratings[developer seniorityLevel]){
        ratings[developer seniorityLevel].push(Object.assign({},
            developer,
            {
                rating : Math.round(skillRating * 100) / 100
            }
        ));
    } else {
        ratings[developer seniorityLevel] = [Object.assign({},
            developer,
            {
                rating : Math.round(skillRating * 100) / 100
            }
        )];
    }
});

Object.keys(ratings).forEach(seniority => {
    ratings[seniority].sort((a, b) => a.rating < b.rating);
    ratings[seniority] = ratings[seniority].slice(0,
projectParams[key].seniority[seniority]);
});

    result[key] = ratings;
});
return result;
}

function prepareWorkerProjectAssignParams(projectId, projectEndDate){
    return {
        updateString : 'set #project = :project',
        expressionAttributeNames: {
            '#project' : 'project'
        },
        expressionAttributeValues: {
            ':project' : {
                projectId,
                endDate : projectEndDate
            }
        }
    }
}

module.exports = {
    prepareGraphAndOrder,
    countDirect,
    countBackward,
    countCriticalPath,
    countFloats,
    countFreeFloat,
    countProjectDuration,
    prepareUpdateStringAndObject,
    countCrash,
    prepareTeam,
    prepareWorkerProjectAssignParams
};

```

**ДОДАТОК Б**

Апробація результатів роботи

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
РАДІОЕЛЕКТРОНІКИ

МАТЕРІАЛИ XXIII МІЖНАРОДНОГО МОЛОДІЖНОГО  
ФОРУМУ

«РАДІОЕЛЕКТРОНІКА ТА МОЛОДЬ  
У XXI СТОЛІТТІ»

16 -- 18 квітня 2019 р.

Том 6

КОНФЕРЕНЦІЯ  
«ІНФОРМАЦІЙНІ ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ»

Харків 2019

23-й Міжнародний молодіжний форум «Радіоелектроніка та молодь у XXI столітті». Зб. матеріалів форуму. Т. 6. – Харків: ХНУРЕ. 2019. – 306 с.

В збірник включені матеріали 23-го Міжнародного молодіжного форуму «Радіоелектроніка та молодь у XXI столітті».

Видання підготовлено факультетом комп'ютерних наук Харківського національного університету радіоелектроніки

61166 Україна, Харків, просп. Науки, 14  
тел./факс: (057) 7021397

E-mail: [mref21@nure.ua](mailto:mref21@nure.ua)

© Харківський  
національний університет  
радіоелектроніки (ХНУРЕ), 2019

## АЛФАВІТНИЙ ПОКАЖЧИК

	D		Горбатенко Б.В.	7
Dehtiarov D.		135	Горбик А.Ю.	233
	H		Гориславец Д.Ю.	216
Honchar Y.		139	Греценко А.В.	299
	P		Грінько К.О.	177
Pydorenko D.O.		115	Губаренко М.С.	200
	R		Гузенко Ю.А.	297
Raniuk V.Y.		54	Гуров А.О.	13
	S		Д	
Shevchenko B.M.		163	Давидова В.П.	74
Stepko A.		121	Демська А.І.	271
	T		Демченко А.Є.	41
Tereshchenko G.U.		52	Деряка Е.В.	25
	A		Джафаров Э.Э.	131
Агарков М.А.		153	Дмитренко А.В.	273
Алешкин А.А.		84	Дмитрієв О.В.	17
Антонюк М.В.		204	Доценко В.В.	245
	Б		Дублищевич Л.А.	109
Байдак В.Е.		127	Є	
Батюченко В.А.		145	Єлецький С.О.	253
Бибичков И.Е.		31	Ж	
Біляєв М.П.		206	Жарков О.Г.	157
Бондарчук А.С.		39	Жернова П.Є.	9
Бояркіна Л. Е.		237	З	
Бредихин Д.В.		78	Зарицкий Д.К.	295
Бронза С.С.		56	Збаражський К.А.	58
	В		Зима А. Е.	220
Вакуленко В.К.		210	Зіміна А.Р.	208
Валковий В.В.		45	И	
Вартанян А.О.		129	Ивановская К.А.	186
Вискребець Д.О.		141	І	
Войтович А.В.		224	Ісаєнко Т.Ю.	291
	Г		К	
Гаевская Д.Ю.		247	Калайда Н.С.	184
Галаган В.В.		21	Карпенко А.В.	257
Глушач Р.В.		11	Карпушенко А.М.	251
Гниденко В.А.		29	Кислинський Є.Ю.	64
Гончаров О.В.		169	Кійко К.В.	43

Клименко Т. Є.	231	Надточій О.Ю.	212
Клюванський Є.Г.	86	Назаренко Ю.В.	72
Коваленко А.И.	106	Никоряк А.В.	249
Ковальова Н. Ю.	149		
Козырь Д.И.	96	О	
Коптев О.О.	50	Огневец Д.В.	263
Корнієць Н.В.	289	Одегова Е.А.	293
Короплясова А.А.	90	Окунева О.Л.	261
Косик Д.О.	165	Ольховський І.В.	188
Косолапов К.С.	35	Ольшанська С.В.	60
Костенко І.М.	229	Орехова І.В.	82
Кочура Л.А.	267	Орленко К.А.	277
Кошевий Д.О.	100		
Кравец М.В.	123	П	
Кудрявцев М.А.	111	Паскарюк Д.О.	198
Кузнецова І.А.	265	Перетятко М.В.	137
Кузьма В.Д.	222	Петрова Е.К.	279
Кузьма Є.А.	68	Петрович Л.С.	109
		Пироженко С.С.	155
Л		Поленок Д.В.	285
Латыш О.С.	269	Попович И.Д.	147
Лизогубов Н.В.	102	Проніна О. І.	182
Лобинцев А.А.	9	Проценко А.А.	194
Лоскутова А.О.	23		
Луганський Є.Д.	171	Р	
Лузан М.О.	15	Редька О.С.	19
Лук'яненко М.В. 7	6	Русаков К.А.	259
Лук'янова К. Ю.	133		
Луніна К.О.	239	С	
Люліна К. П.	161	Светличная В.Г.	287
		Сербіна Д.В.	45
М		Сидорчук Е.И.	202
Маєсенко В.С.	80	Синкевич М.Э.	151
Махія І.М.	94	Сідельник В.О.	74
Мигалина Ю.І.	180	Скакун Р.Г.	98
Минаков С.А.	88	Скляр В.О.	92
Михневич Т.К.	113	Слепанская В.Д.	33
Міллер П.П.	117	Слива Є.К.	218
Міхневич І.Д.	119	Сокол В.В.	31
		Спасибін М.Р.	226
Н		Стрименешенко А.С.	190, 192
Нагорна Я. А.	231	Стьопін В.І.	7
		Сушинський І.К.	173

## ІНСТРУМЕНТИ ПЛАНУВАННЯ ДО ЗВІТНОСТІ, АБО ЯКІ ПРОГРАМНІ ДОДАТКИ КРАЩЕ ВИКОРИСТОВУВАТИ В ПРОЕКТНОМУ МЕНЕДЖМЕНТІ

Лук'янова К. Ю.

Науковий керівник – проф. Дудар З. В.

Харківський національний університет радіоелектроніки  
(61166, Харків, просп. Науки, 14, каф. Програмної Інженерії,  
тел. (057) 702-14-46)

E-mail: [kateryna.lukianova@nure.ua](mailto:kateryna.lukianova@nure.ua), тел. +380951481457

The given work is devoted to the analysis of modern project management applications that optimize planning, monitoring and control processes in information technology and software products development. A Gantt chart, Product Breakdown Structure and Work Breakdown Structure are standard techniques used in project planning that can be automated. In this abstract, we have compiled a list of extremely effective team management and project planning tools, that can help to optimize process to plan essential project milestones, and that you can adopt today and be more efficient in project planning and team coordination.

Упродовж останнього десятиліття управління проектами стає все більш складним. Через цю складність при виконанні великих проектів - перевищення термінів, в інформаційних технологіях - часто запізнюються строки та перевищується бюджет. Управління ІТ-проектами включає в себе нагляд за розробкою програмного забезпечення, інсталяцією обладнання, регулярним оновленням, підтримкою хмарних обчислень, проектами аналізу та впровадженням ІТ-послуг. На додаток до звичайних проблем, які можуть призвести до провалу проекту, є фактори, які також можуть негативно вплинути на успіх ІТ-проекту:

- зміни в технології на стадії впровадження проекту;
- інфраструктурні зміни, які впливають на безпеку і управління даними;
- невідомі залежні відносини між обладнанням, програмним забезпеченням мережевої інфраструктури та даних.

ІТ-проекти також можуть не бути завершені в строк, якщо вибрана технологія розробки вперше застосовується. Це загальний ризик, тому що нова технологія не була реалізована або використана раніше в організації, і, відповідно, виникнуть ускладнення, які вплинуть на успіх проекту. Саме тому наявність програмного забезпечення для управління проектами відіграє ключову роль в успішності проекту. Програмне забезпечення для управління проектами - це програмне забезпечення, яке допомагає керівникам проектів (PM) і командам співпрацювати і досягати цілей у спільний спосіб, керуючи ресурсами та витратами. Функції можуть включати розподіл завдань, відстеження часу, складання бюджету, планування

ресурсів за допомогою діаграм Ганта, колективне співробітництво та багато іншого. Подібним чином, компанії сьогодні можуть краще спланувати роботу, виявивши ризикові аспекти проекту, використовуючи програмне забезпечення для управління проектами.

Метою даної роботи є аналіз сучасних програмних додатків, які оптимізують процес планування, контролювання проекту та координування команди. Нижче наведена таблиця з порівняльними характеристиками найпоширеніших програмних додатків у плануванні проектів на сьогодні:

Таблиця. Порівняльна характеристика програмних додатків

	Smart Sheet	Gantt Pro [2]	Asana	Wrike[3]	MS Project
Управління завданнями		✓	✓		✓
Управління ресурсами		✓	✓	✓	✓
Діаграми Ганта	✓	✓		✓	✓
Планування розкладу		✓		✓	✓
Підтримка Agile			✓	✓	✓
Підтримка Waterfall	✓	✓	✓		✓
Облік робочого часу	✓	✓		✓	
Управління бюджетом	✓		✓		✓
Зручність згідно Capterra [1]	4.5(749)	4.5(65)	4.5(5797)	4(1193)	4.5(1811)
Free Trial	умовно безкоштовна	14 днів повнофункціональна	умовно безкоштовна	обмежена	умовно безкоштовна

Комплексні програмні додатки для управління проектами, такі як п'ять згаданих вище, мають можливість автоматично пов'язувати інформацію про завдання, графіки та ресурси в діаграмах Ганта з іншими даними такими, як процеси та час, щоб дати більш точну, реальну та високорівневу картину успіху проектів. На сьогодні найкращими додатками для управління проектами являються GanttPro та Microsoft Project.

#### Список літератури:

1. Project management software [Електронний ресурс] // Режим доступу: <https://www.capterra.com/project-management-software/>.
2. Project management solution based on Gantt chart [Електронний ресурс] // Режим доступу: <https://ganttpro.com/>
3. Real-time Work Management Software [Електронний ресурс] // Режим доступу: <https://www.wrike.com/gantt-chart/>.

## ДОДАТОК В

### Слайди презентації

Атестаційна робота магістра

## Методи оцінювання ефективності програмних проектів та оптимізації процесів розробки

Виконала:

ст. гр. ІПЗм-17-1

Лук'янова К.Ю.

Керівник роботи:

проф. Дудар З.В.

## Мета роботи

- ▶ Дослідження методів оцінки ефективності програмних проектів;
- ▶ Проведення опитування українського ІТ-ринку для формування емпіричних знань про характеристики успішного керування проектами;
- ▶ Створення WEB-додатку, який допомагає менеджерам проекту оптимізувати процес розробки проекту.

## Аналіз стану досліджень

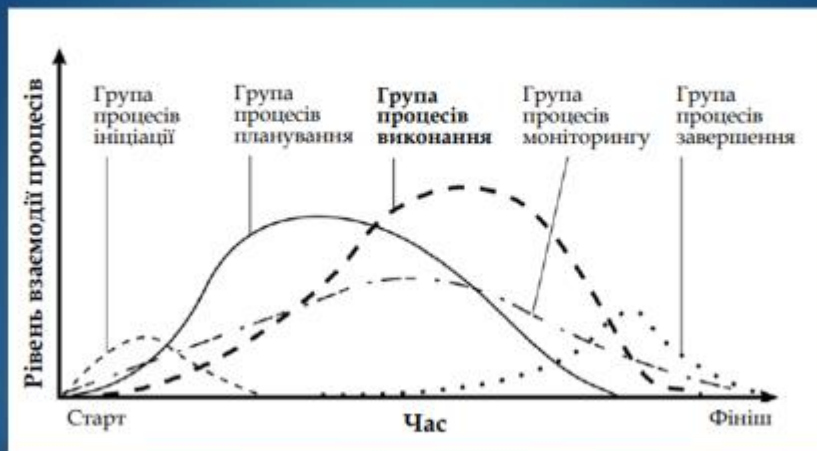
MODERN RESOLUTION FOR ALL PROJECTS					
	2011	2012	2013	2014	2015
<b>SUCCESSFUL</b>	29%	27%	31%	28%	29%
<b>CHALLENGED</b>	49%	56%	50%	55%	52%
<b>FAILED</b>	22%	17%	19%	17%	19%

The Modern Resolution (OnTime, OnBudget, with a satisfactory result) of all software projects from FY2011 - 2015 within the new CHAOS database. Please note that for the rest of this report CHAOS Resolution will refer to the Modern Resolution definition not the Traditional Resolution definition.

## Методи оцінювання ефективності керування програмними проектами

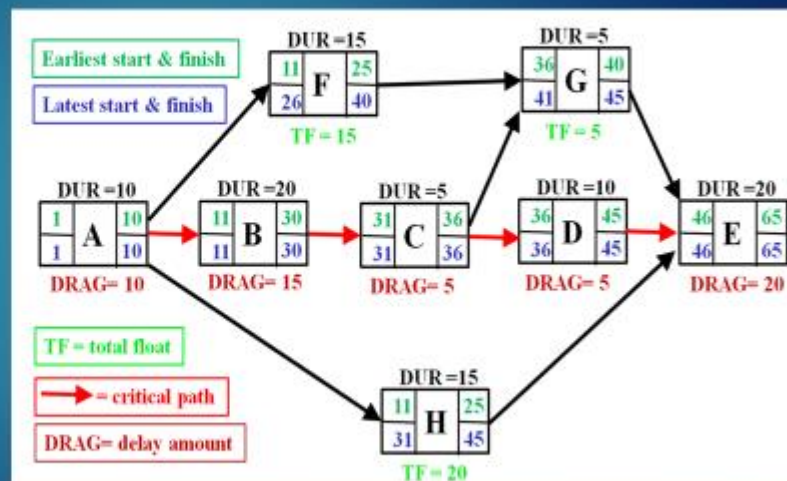
- ▶ на основі анкети
- ▶ на основі метрик
- ▶ на основі моделі

## Процеси управління IT-проектом



## Метод критичного шляху

- ▶ визначити задачі;
- ▶ опрацювати їх послідовність за допомогою таблиці;
- ▶ скористатися діаграмою на манер, який зображено та заповнити отриманими даними;
- ▶ оцінити час, який необхідно буде витратити на кожну із задач;



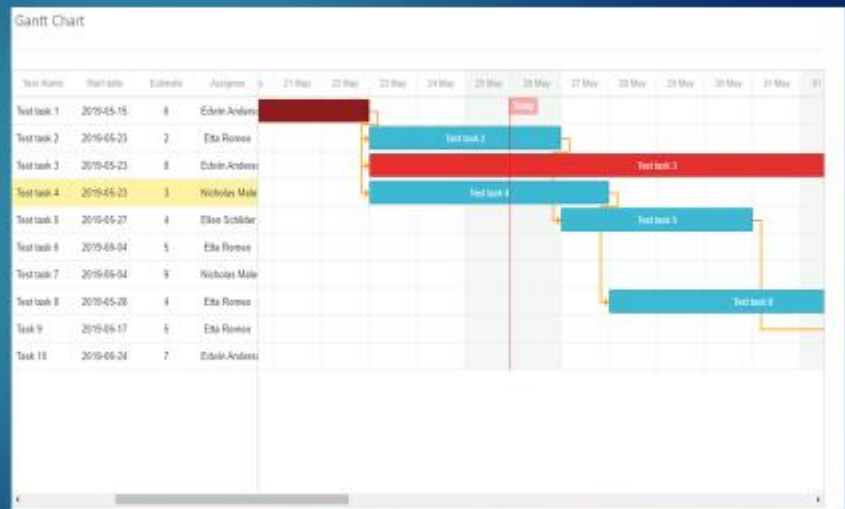
## Діаграма Ганта

Діаграма Ганта являє собою відрізки, розміщені на горизонтальній шкалі часу.

Кожен відрізок відповідає окремому завданню або підзадачі. Завдання і підзадачі, складові плану, розміщуються по вертикалі.

Початок, кінець і довжина відрізка на шкалі часу відповідають початку, кінцю і тривалості завдання.

На деяких діаграмах Ганта також показується залежність між завданнями.

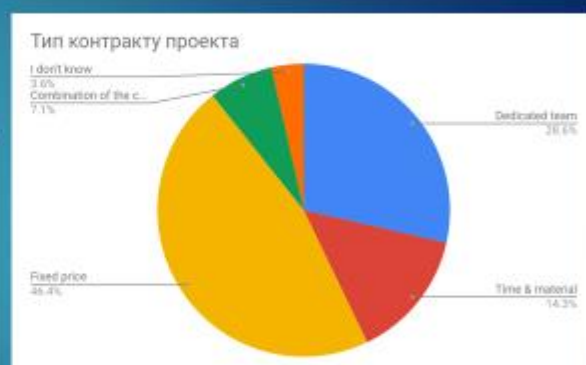
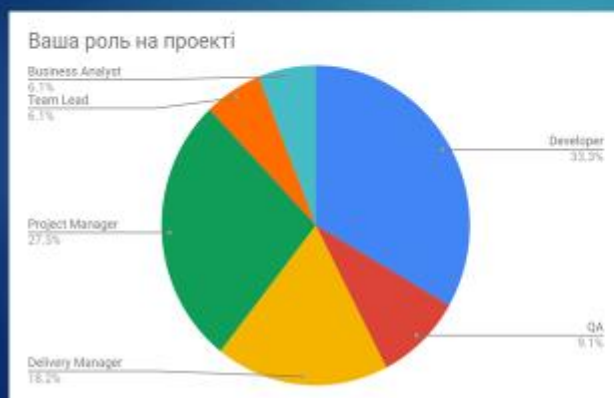


## Мета проведення опитування українського ІТ-ринку

- ▶ залежність успішності проекту від типу контракту (fixed-price, time and material, dedicated team);
- ▶ як розмір проекту та методологія управління впливають на успішність проекту;
- ▶ який з етапів управління проектами найбільш ризиковий
- ▶ які фактори часто забувають оцінити на етапі планування проекту

## Аналіз отриманих результатів

- 81 респондент
- 22 IT компанії



## Результати опитування щодо успішності проекту

	5 (відмінно)	4	3	2	1
Проект розроблено в межах бюджету	28%	24%	40%	8%	0%
Проект розроблено вчасно	30%	28%	28%	14%	0%
Якість проекту	53%	35%	12%	0%	0%
Розроблена функціональність співпадає з вимогами клієнта	45%	39%	11%	5%	0%

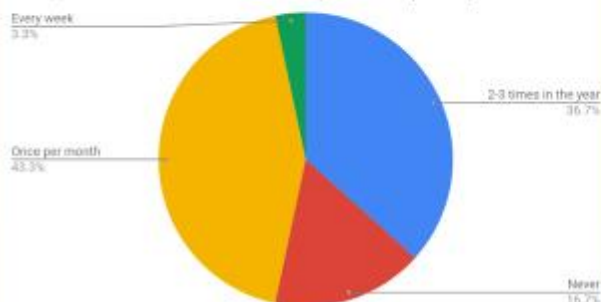
## Типи завдань, які найчастіше забувають врахувати на етапі планування ІТ-проекту

- ▶ час, на адаптацію в проєкті для нових співробітників;
- ▶ налаштування необхідного програмного забезпечення та отримання облікових даних;
- ▶ розгортання проєкту на виробництві;
- ▶ уточнення вимог проєкту;
- ▶ покращення продуктивності розробленого продукту;
- ▶ проєктні комунікації в команді та зустрічі з замовником;
- ▶ стабілізація проєкту та виправлення дефектів;
- ▶ передача проєктних знань;
- ▶ створення технічної документації.

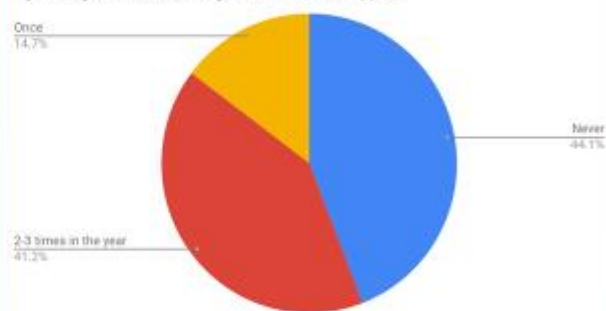


## Інфографіка роботи у вихідні дні

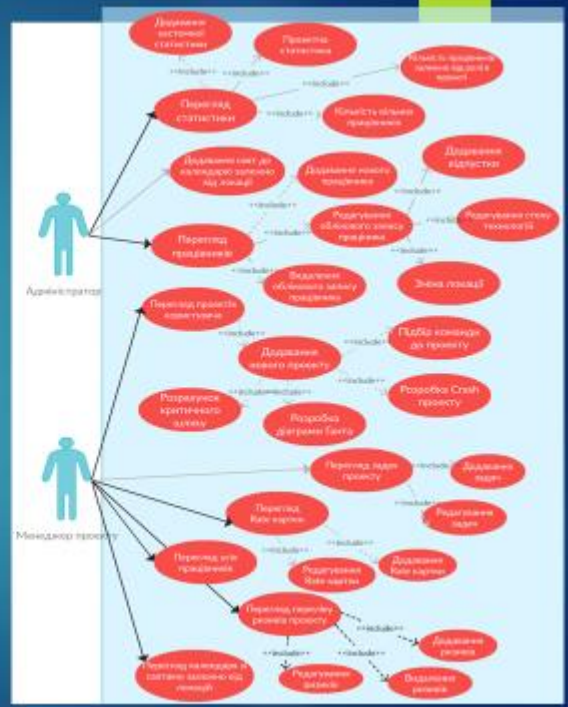
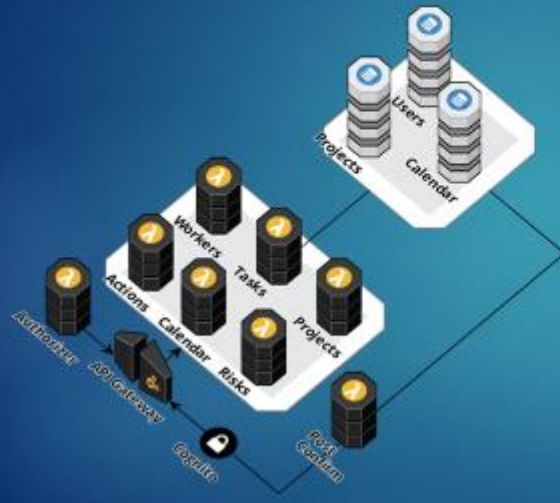
Як часто Вам потрібно було залишатися переробляти у вихідні за наказом клієнта/керівника проєкту?



Як часто Ви виходили на роботу за наказом керівника проєкту або клієнта у святкові вихідні?

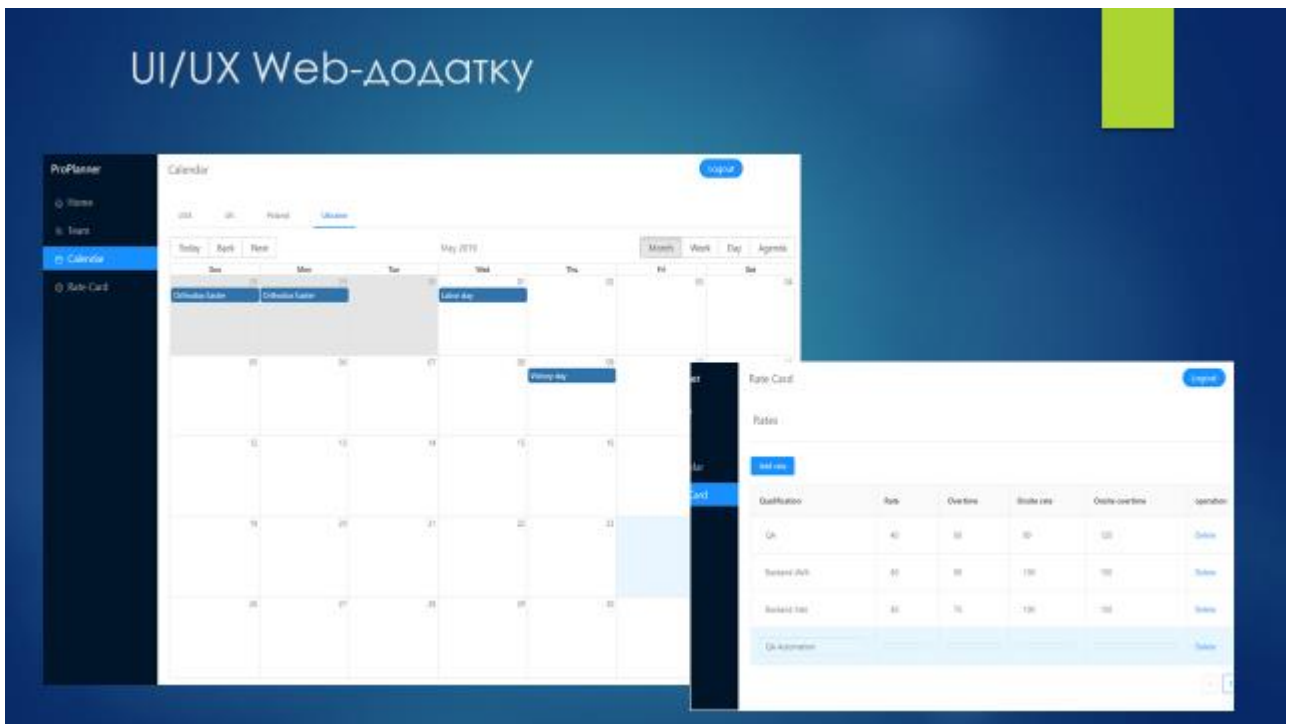
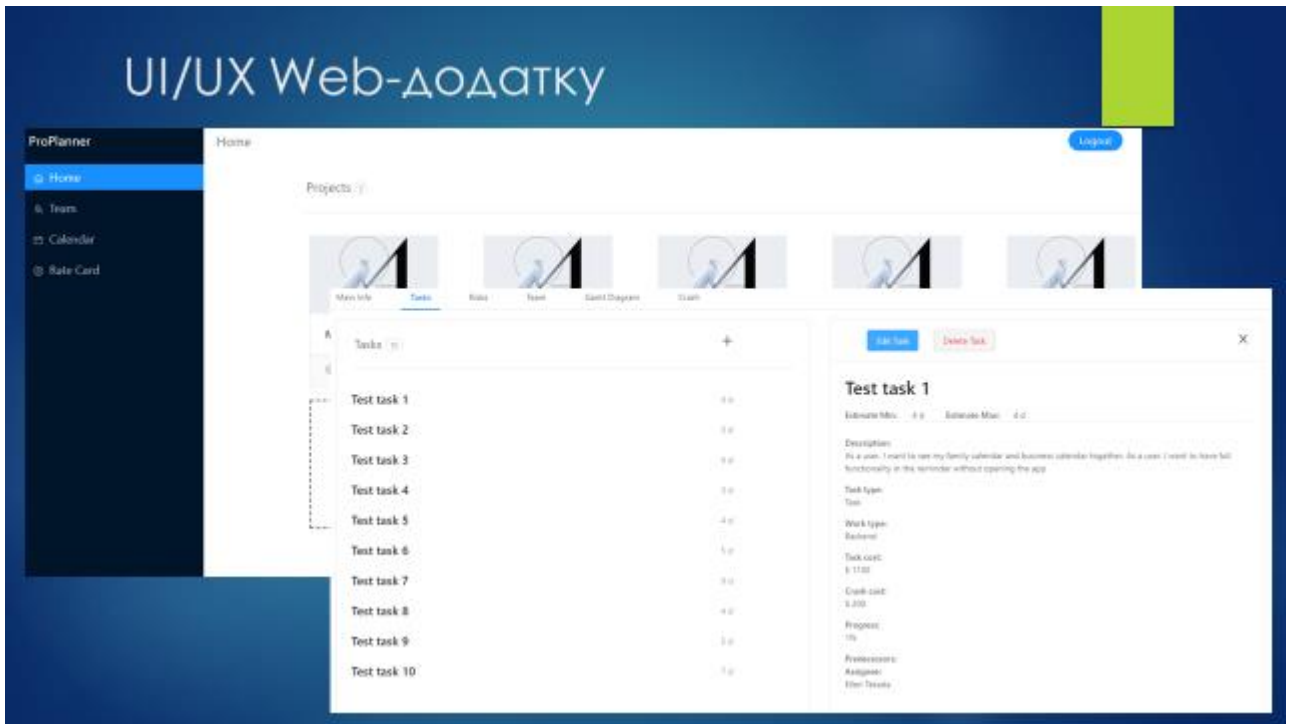


# Реалізація Web-додатку



# Вибір технологій та засобів реалізації





## Сторінка пошуку команди на проект

**Create team**

\* Team type: Backend | \* Main skill: Android | \* Skills: OOP x, Android SDK x, TDD x, Android Studio x, JSDC x

\* Senior: 1 | \* Middle: 0 | \* Junior: 2

\* Team type: Frontend | \* Main skill: JavaScript | \* Skills: React x, HTML x, CSS x, Responsive design x, UI x, UX x

\* Senior: 0 | \* Middle: 2 | \* Junior: 1

+ Add team | Cancel | Create team

**Team**

Default | Middle | Small

Name	Main skill	Seniority level	Location	Rating
Erna Cantini	Java	Senior	UK	0.5
Charles Robbins	Java	Middle	Ukraine	0.5
Eta Romeo	Java	Middle	UK	0.33
Cory Bertrand	JavaScript	Senior	UK	0.67
Elen Schilder	JavaScript	Middle	USA	0.67
Gda Sans	JavaScript	Middle	USA	0.5

Cancel | Create team

## Сторінка розрахунку скорочення термінів проекту

**Gantt Chart**

Task name	Start date	End date	Assignee
Test task 1	2019-05-15	6	Eduin Andros
Test task 2	2019-05-23	2	Eta Romeo
Test task 3	2019-05-23	8	Eduin Andros
Test task 4	2019-05-23	3	Nicholas Mair
Test task 5	2019-05-27	4	Elen Schilder
Test task 6	2019-05-28	5	Eta Romeo
Test task 7	2019-06-04	9	Nicholas Mair
Test task 8	2019-05-25	4	Eta Romeo
Test 9	2019-06-17	6	Eta Romeo
Test 10	2019-06-28	7	Eduin Andros

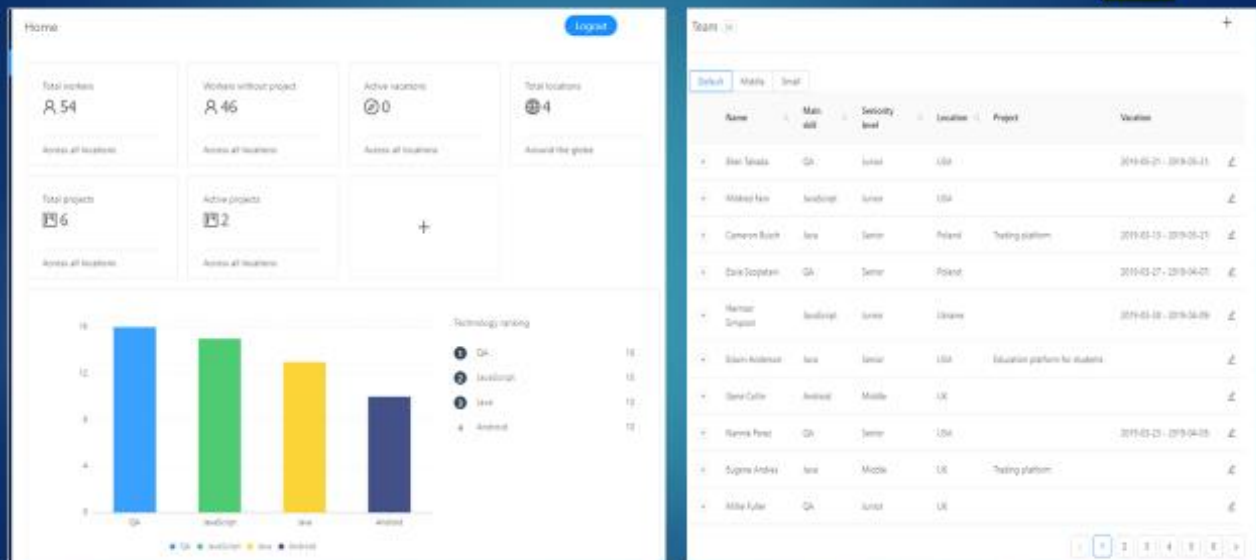
**Project Gantt in Crash**  
Planned End Date - 3 Jul 2019 | Crash End Date - 5 Jul 2019

Task name	Start date	End date	Assignee
Test task 1	2019-05-15	4	Eduin Andros
Test task 2	2019-05-21	2	Eta Romeo
Test task 3	2019-05-21	8	Eduin Andros
Test task 4	2019-05-21	3	Nicholas Mair
Test task 5	2019-05-23	4	Elen Schilder
Test task 6	2019-05-27	5	Eta Romeo
Test task 7	2019-06-31	9	Nicholas Mair
Test task 8	2019-05-24	4	Eta Romeo
Test 9	2019-05-13	5	Eta Romeo
Test 10	2019-06-28	7	Eduin Andros

Crash ID	Cost	Save	Project Duration	Project Cost	Test task 1	Test task 2	Test task 3	Test task 4
Initial	—	—	35	25400	5	2	8	3
Crash 1	200	100	34	25300	5	2	8	3
Crash 2	200	100	33	25200	4	2	8	3
Crash 3	200	100	32	25100	4	2	8	3
Crash 4	200	100	31	25000	4	2	8	3

Cancel | Apply

## Адмін панель



## Використання результатів роботи в майбутньому

Переваги додатку серед аналогів:

- ▶ алгоритм автоматичного аналізу та пошуку ресурсів на проект;
- ▶ розрахунок критичного шляху проекту;
- ▶ врахування до тривалості проекту святкових вихідних днів, у тих локаціях, де знаходиться команда;
- ▶ аналіз вигідності зменшення тривалості проекту, та як це впливає на бюджет проекту
- ▶ гнучкість та масштабованість системи

Подальший розвиток:

- ▶ Інтеграція з JIRA
- ▶ Додати можливість ведення документообороту
- ▶ Додати доступ співробітникам для заповнення реального часу витраченого на завдання
- ▶ Розробка алгоритму для врахування часу на стабілізацію проекту

## Висновки

У ході виконання магістерської роботи було проведено:

- ▶ Аналіз стану дослідження;
- ▶ Описано проведено дослідження українського ІТ-ринку;
- ▶ Розроблено Веб-додаток;
- ▶ Розглянуто подальше використання отриманих результатів в науковій та практичній діяльності.