ONE INSTANCE OF AN 'ADVANCED TOPICS IN SOFTWARE ENGINEERING' MASTER COURSE

NATAŠA SUKUR

DEPARTMENT OF INFORMATICS AND MATHEMATICS

FACULTY OF SCIENCES

UNIVERSITY OF NOVI SAD, SERBIA

16TH WORKSHOP ON "SOFTWARE ENGINEERING EDUCATION AND REVERSE ENGINEERING", JAHORINA 22ND – 26TH AUGUST 2016

OVERVIEW

- INTRODUCTION
- ADVANCED TOPICS IN SOFTWARE ENGINEERING COURSE
- RESULTS
- CONCLUSIONS

INFORMATICS MASTER STUDIES

- The main goal of informatics master studies is education of experts in the field of Informatics in three study programs
  - **Computer Science**
  - **Information Technologies**
    - **Software Engineering**
    - **Information Systems**
  - **Teaching Methods in Informatics**
- The courses are designed to accomplish the following
  - Enable students for individual and team work
  - Software systems development
  - Application of formal methods
  - ...

ELECTIVE COURSES

Предмет	Фонд ч.	ЕСПБ
Информатички пројекат	2+4+3	10
Примењена статистика	2+3	7
Конструкција компајлера 1	2+2+1	7
Теорија графова	2+2	6
Математичка логика у рачунарству	2+2	6
Семинарски рад Ц	1+0+3	6
Еволуција софтвера	2+1+2	7,5
Развој заснован на компонентама	2+1+2	7,5
Формални методи у инжењерству	2+1+2	7,5
Напредне теме софтверског инжењерства	2+1+2	7,5
Комбинаторни алгоритми	2+2+1	7
Рачунарска графика 2	2+1+2	7
Историја информатике	2	4
Софт. инжењерство за системе база података	3+1+1	7,5
Паралелно програмирање	3+3	8
Интеграција система	3+1+2	8
Инжењерство захтева	3+1+1	7,5
Социјалне мреже	2+1+1	7
Управљање софтверским пројектима	3+2+1	8
Тестирање софтвера	2+1+2	7,5
Софт. инжењерство у критичним системима	2+1+2	7,5
Архитектура, дизајн и обрасци	3+1+1	7,5
Дистрибуирани системи	2+1+2	7,5
Семинарски рад Д	1+0+3	6
Конструкција компајлера 2	2+1+2	7
Процес развоја информационих система	3+1+1	7,5
Структуре података и алгоритми 3	2+2+1	7
Вештачка интелигенција 2	2+1+2	7

Only a few elective courses over the year include doing research

- Seminary C
- Seminary D
- **Advanced Topics in Software Engineering**

ADVANCED TOPICS IN SOFTWARE ENGINEERING

- Made for students interested in doing research
- Work on a complex project
- Individual or team work
- Demands cooperation and communication

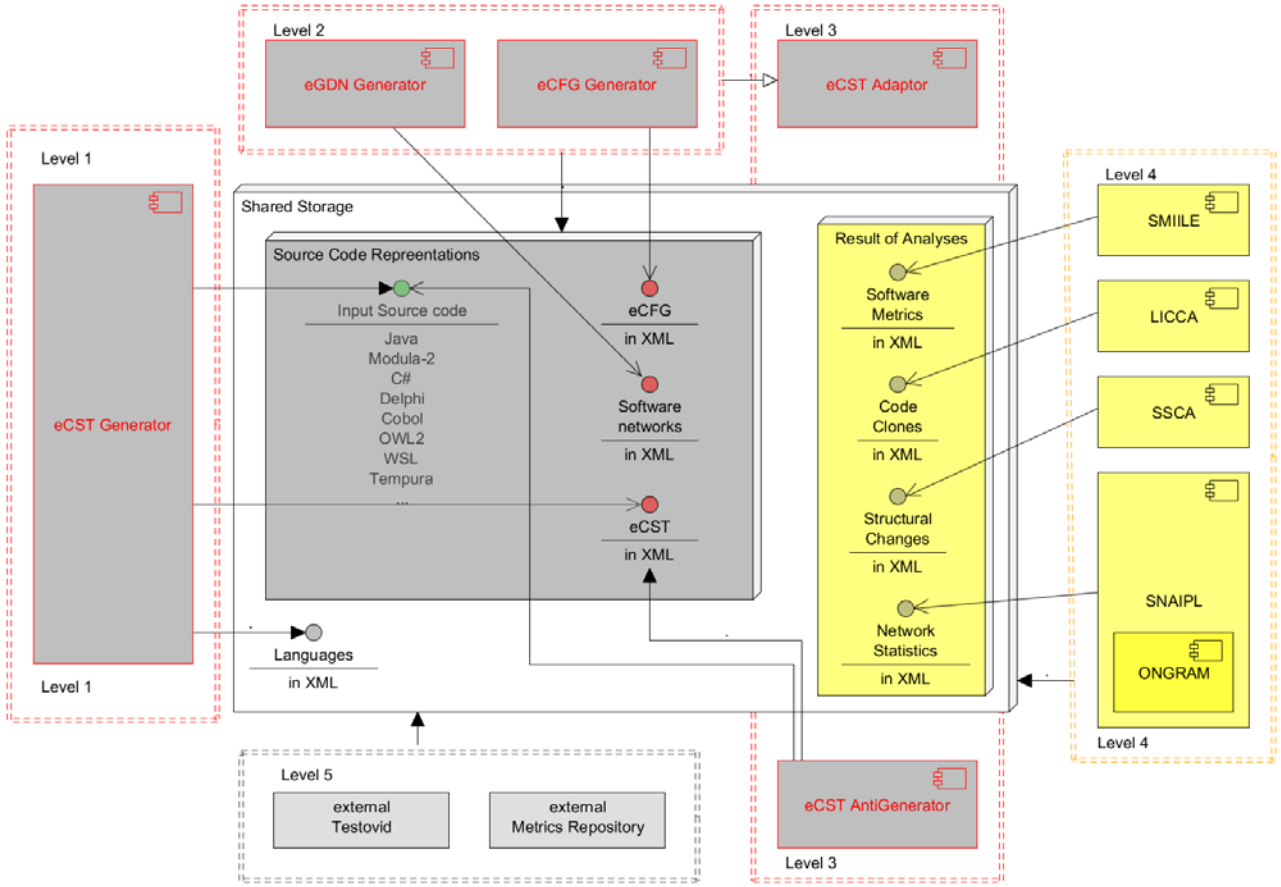
- Our work:
  - Three ambitious master students (all PhD students starting from October) working in a team
  - Work based on Worst Case Execution Time in SSQSA framework

SSQSA FRAMEWORK

- **SSQSA = Set of Software Quality Static Analyzers**

- Aim
  - to provide a platform for building set of static software analyzers to ensure, check and consequently increase the quality of software products
- Characteristic
  - language independency based on eCST (enriched Concrete Syntax Tree) representation of source code
  - Nodes
    - **universal nodes** with predefined, language-independent meanings which denote semantic concepts expressed by constructs of the language
    - tokens, elements of the source code, which are leaf nodes of eCSTs. Important characteristic is that every single symbol existing in the source code has to appear in leafs of the corresponding eCST. [Rakić, 2015]

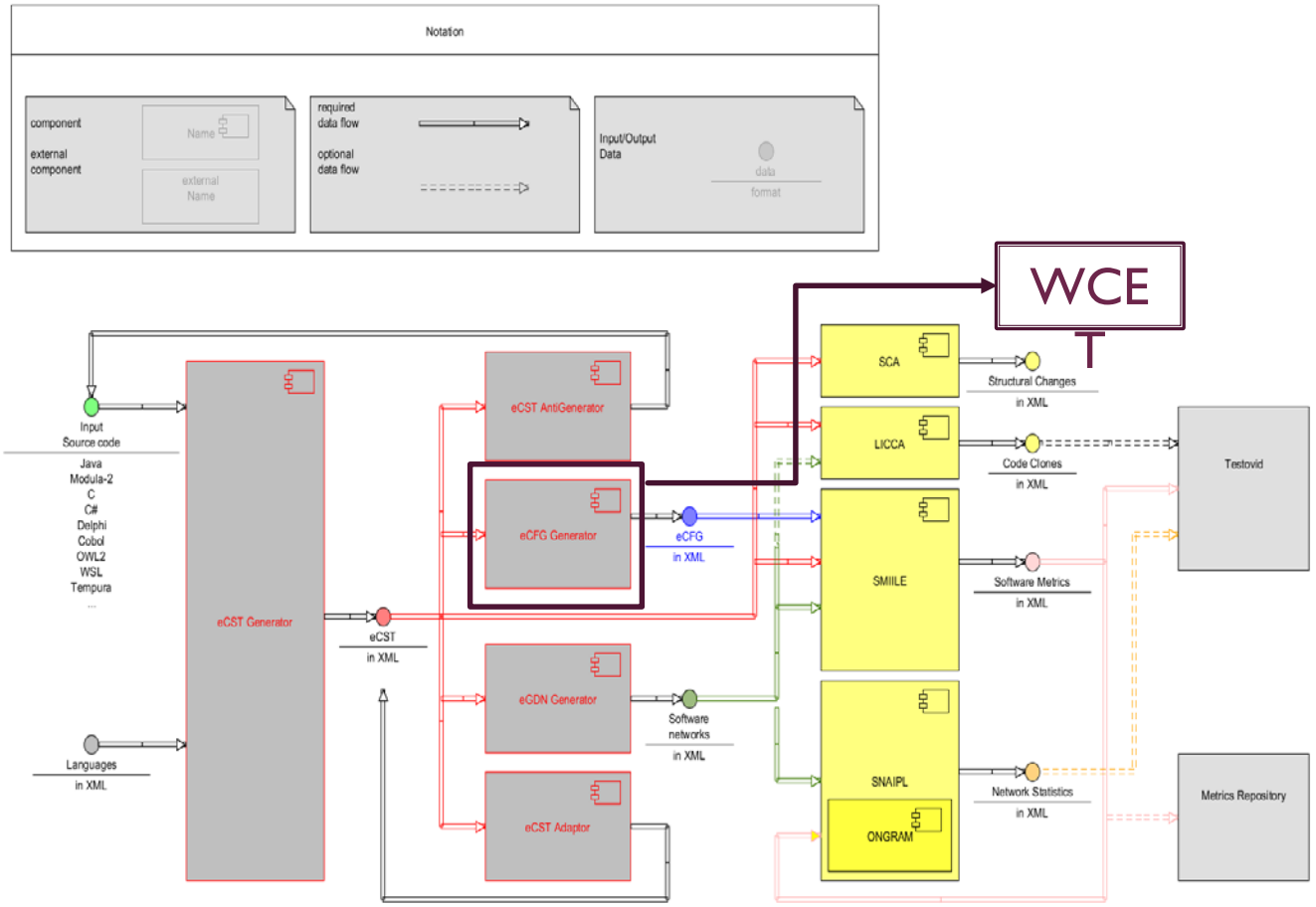
SSQSA ARCHITECTURE



WORST CASE EXECUTION TIME

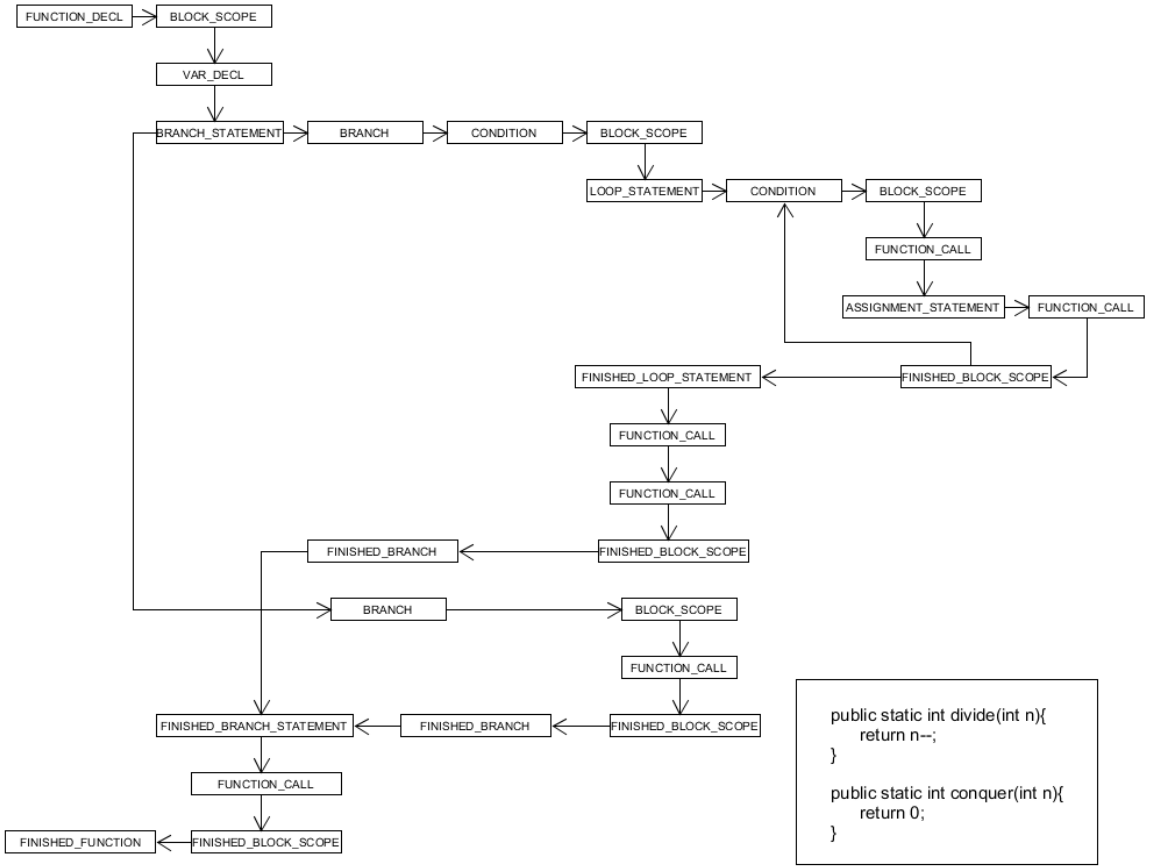
- Part of timing analysis
- Represents the upper (worst) value of the execution time of a program
- Importance
  - Real-time systems must satisfy some time limitations
- WCET estimation on source code level
  - Allows us to sometimes perform this estimation in the earliest phases of software development
  - Otherwise, the result could be a highly costly system re-design [Gustafsson et al. 2009]

ADDING WCET ESTIMATION TO SSQSA USING CONTROL FLOW GRAPHS



RESULTS

```
public static void divideAndConquer() {

    int n = 100;

    if (n > 13) {

        while(n > 13) {

            System.out.println("Dividing n");

            n = divide(n);

        }

        System.out.println("n is conquerable");

        conquer(n);

    }

    else {

        conquer(n);

    }

    System.out.println("n is now 0");

}
```




```
public static int divide(int n){
    return n--;
}

public static int conquer(int n){
    return 0;
}
```

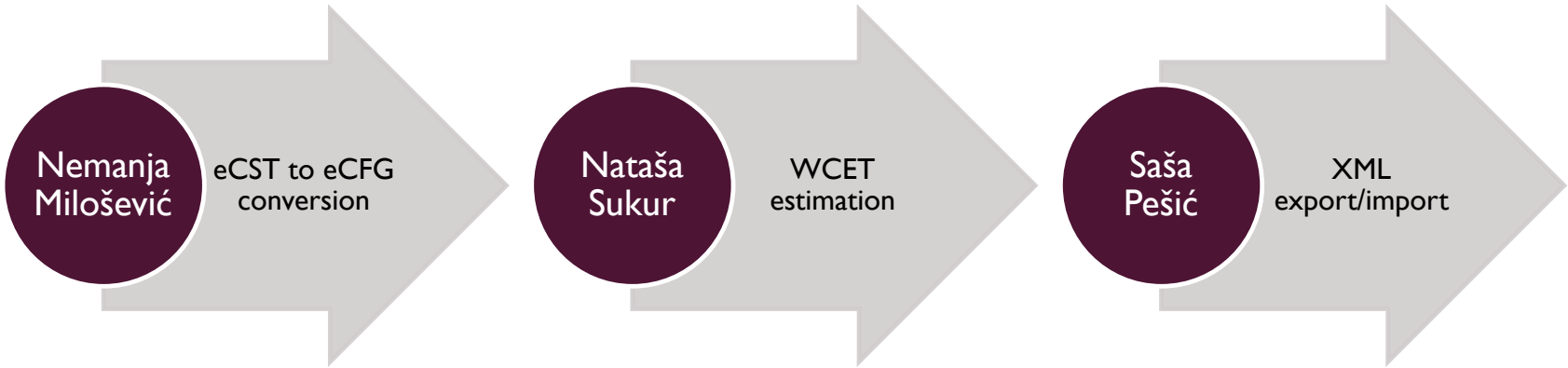
THE IMPORTANCE OF WCET INTEGRATION TO SSQSA

- Precise, language independent WCET estimation
- Indicate the irregularities in the phase of writing the source code
- Estimation rich in detail, sometimes invisible to dynamic WCET estimation
  - All useful data from the source code is extracted
  - eCST intermediate structure is constantly improving in order to provide all the necessary data for successful and precise analyses
- Goal: as successful estimation as possible, no matter the input language

PARTICIPANTS

dr Zoran Budimac	top level decisions
dr Gordana Rakić	high level decisions communication coordination
Jozef Kolek	counselling testing
Nataša Sukur Nemanja Milošević Saša Pešić	implementation

WORK DIVISION IN IMPLEMENTATION



ORGANIZATION

- Meetings
  - Crucial problem solving
  - Decision making on the approach
  - Not very often, usually done through online communication
- Implementation done in coordination with supervisors
  - Not strictly determined and controlled
  - Freedom in problem solving

WHAT KEPT US GOING VS. WHAT SLOWED US DOWN

- What kept us going
  - Doing research
  - Getting to know complex projects
  - Working with more experienced people
  - Contributing to a meaningful project
  - Acknowledgement

- What slowed us down
  - A lot of communication required
  - Not knowing the SSQSA project well enough
  - Selecting approach in problem solving

FINAL RESULTS

- **First results of WCET estimation in SSQSA framework**
  - Accepted scientific paper for SQAMIA 2016 workshop in Budapest

- Results and benefits
  - Applying the knowledge gained from Research methods course
  - Getting to know the „language“ of scientific papers
  - Questioning our results
  - Doing research on related work
  - Planning future work based on gained knowledge

CONCLUSIONS ON ATSE COURSE

- Encourages young students to learn more
- Introduces students to complex projects
- Gives insight into process of doing research
- Gives good basis for future researchers
- Good proof whether the students' gained skills are enough for doing and understanding this kind of work

REFERENCES

- J. Gustafsson, P. Altenbernd, A. Ermedahl, B. Lisper. 2009. Approximate Worst-Case Execution Time Analysis for Early Stage Embedded Systems Development. Proc. of the Seventh IFIP Workshop on Software Technologies for Future Embedded and Ubiquitous Systems (SEUS 2009). Lecture Notes in Computer Science (LNCS), Springer, pp. 308-319
- G. Rakić. 2015. Extendable and Adaptable Framework for Input Language Independent Static Analysis, Novi Sad Faculty of Sciences, University of Novi Sad, doctoral dissertation, p. 242

THANK YOU FOR YOUR ATTENTION ☺

QUESTIONS?

natasa.sukur@uns.ac.rs