Study of Task Processes for Improving Programmer Productivity

Damodaram Kamma
Indraprastha Institute of Information Technology, Delhi
Thesis Advisor: Prof. Pankaj Jalote

Software Process & Productivity

- Significant improvement in overall software process for improving productivity
- Majority of effort in a project is spent at task level by programmers
- In a given overall software process, software productivity depends on how efficiently each programmer execute their assigned tasks

Task Processes

- Programmers execute a task incrementally in small steps
- Organization of these steps is called a “Task process”
- Overall software process do not standardize any task process
- Task processes may vary from Programmer to Programmer and have an effect on productivity

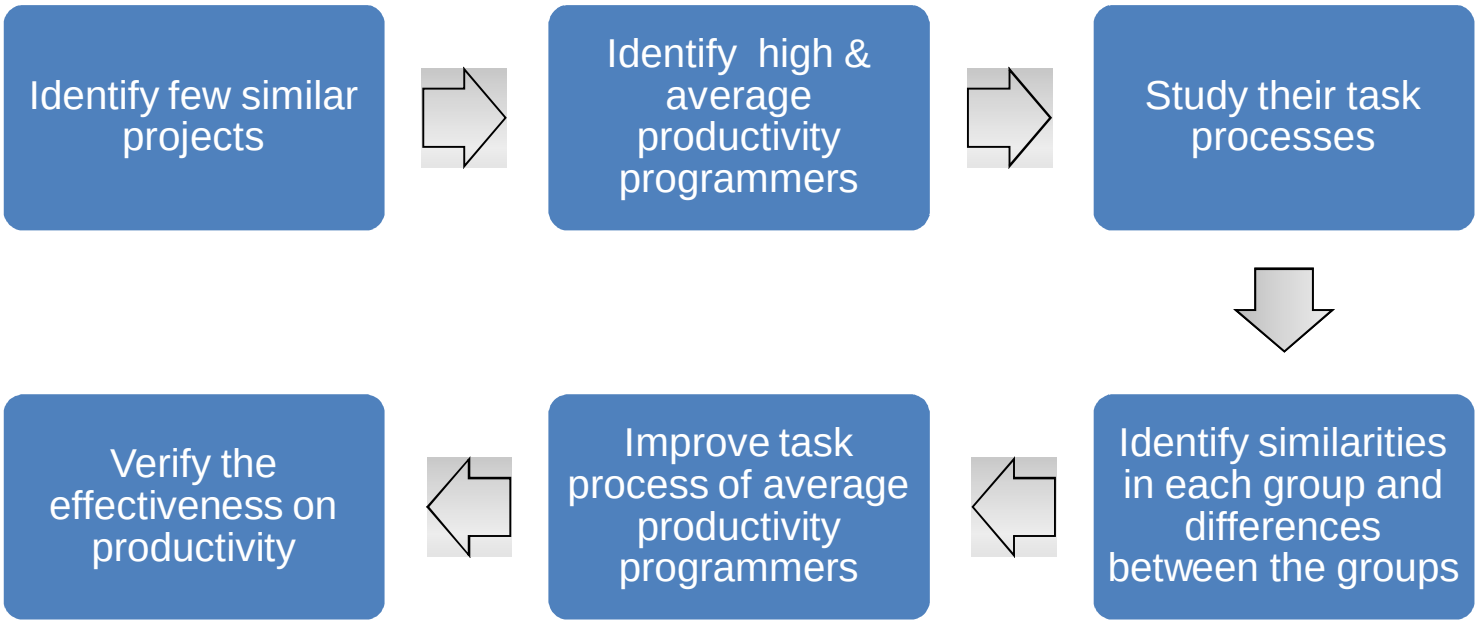
Research Questions

- Do programmers have a dedicated task process?
- What are the task processes used by programmers?
- Do task processes affect productivity?
- Which task processes have positive impact and which ones have a negative impact on programmer productivity?
- Can we utilize productive task processes to improve productivity?

Related work

- Scientific Management Theory
  - Few people are more productive than others
  - There is always a best way to perform any type of task
  - Possible to improve productivity of average individuals
- Industrial Engineering
  - "Standardized work" and "Value stream analysis" focus on task process
- Personal Software Process
  - Improves programmer productivity
  - Did not concentrate on studying and understanding task processes between programmers

Approach


Identify few similar projects
Identify high & average productivity programmers
Study their task processes
Identify similarities in each group and differences between the groups
Improve task process of average productivity programmers
Verify the effectiveness on productivity

Field Study

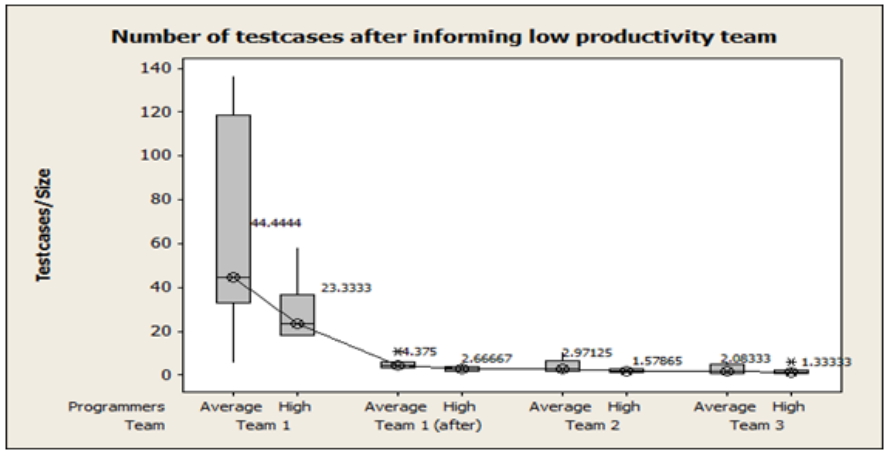
- Three unit testing projects in model based software development
- Programmers video captured their computer monitors
- Manually analyzed task videos

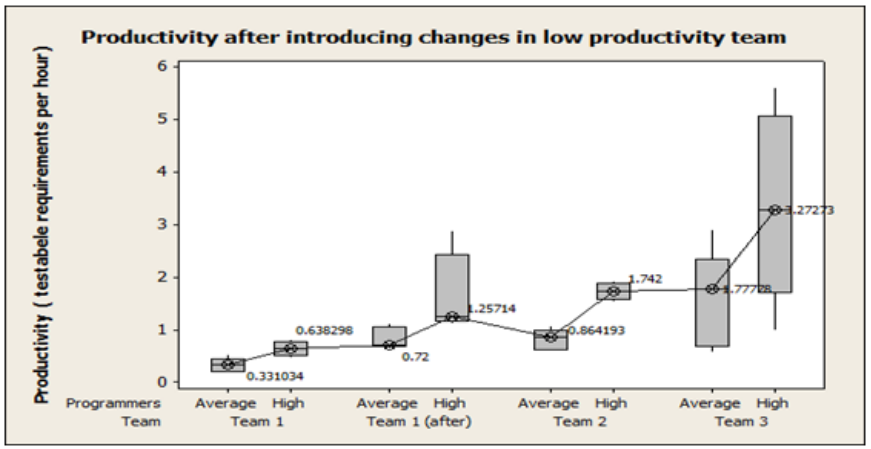
Preliminary results

- Each programmer has a dedicated task process
- High productivity programmers have similar task processes
- Task processes of average productivity programmers vary and are different from high productivity programmers
- Few differences in task processes include
  - Automation used by high productivity programmers
  - Way of deriving test cases
  - Number of test cases and test executions
  - Usage of tools to comprehend the given task

Preliminary results

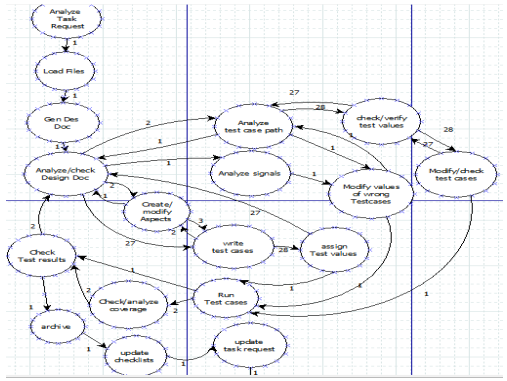
- Productivity differences observed between projects too
- Low productivity team was informed about few task processes
  - Automation in high productivity teams
  - Too many test cases in low productivity team





Challenges and Future work

- Formal representation & comparison of task processes
  - ❖ Structure of task process
    - Organization of various steps
      - o Simple flowcharts/activity diagrams?
      - o Discrete time Markov models?
  - ❖ Quantitative and qualitative attributes associated with each step?



Challenges and Future work

- Transferring productive task processes to average productivity programmers

- Overall effectiveness of this approach on productivity
  - Need to capture data multiple times at different time points
  - Stability of projects – programmers, tools, processes?
  - Calculating productivity and linking with task processes?

Thank you