

Software Multi-Project Resource Scheduling: A Comparative Analysis

Fei Dong, Mingshu Li, Yuzhu Zhao, Juan Li, Ye Yang

Laboratory for Internet Software Technologies

Institute of Software , Chinese Academy of Sciences

Agenda

- Motivation
- State of the art
- A Comparison Framework
- Comparison Results and Analysis
- Conclusion and Future Work

Motivation

- Why software multi-project management?
 - Software organizations commonly work in multi-project environments
 - Software project has unique characters, and experiences in other industries can not be borrowed mechanically
- Why resource scheduling?
 - Human resource usually represents the major cost of software projects, and its scheduling is really important as well as difficult.

Challenges of resource scheduling

- Unstable labor demand along the project life cycles
 - Staffing projects in a lump at their beginning may lead to either insufficiency or surplus in manpower.
 - It is especially true for software development project with high uncertainty
- High dependence on knowledge or skills
 - require assigning right people right task so as to increase productivity and product quality
 - facilitate the necessary communication and minimize its cost
- Increasing concern about humanity

Asking for more attention and researches...

- Little systematic study concentrates on managing multiple software projects
- Little research about multi-project comes from software development community.

<i>Source</i>	<i>Total</i>	<i>With author(s) from software/CS/IS community</i>	<i>%</i>
IEEE	219	45	20.5
IJPM	78	3	3.8

– Multi-project/program/portfolio, not adding resource keywords

Literature survey

- Key words search.
 - An Internet search via Google Scholar was conducted.
 - To obtain practices in software area, IEEE and ACM database were searched particularly.
- Classical or authoritative materials
 - e.g. PMI standards
- Web forum
 - There is little research paper about agile multi-PM, while indeed there are some explorations in industry. We browsed famous forums where agile practitioners gather and also asked active posters.

State of the art

- One project at a time
- Networking optimization/heuristics algorithms
- Critical chain methodology
- Resource modeling /simulation
- Multi-agent scheduling
- Scrum of scrums /Virtual team
- Fixed core team with flexible resource transfer
- Pair programming across multiple projects
- Exchange by turns
- Multi-product team
- Classify resources and schedule with different mechanisms

A Comparison Framework

- Value orientation
- Centralization
- Homogeneity
- Complexity
- Uncertainty
- Executive ability

Value orientation

- Goal of resource scheduling or management.
 - More emphasis on product innovation and quality, or resource utilization and cost?
- Attitude to the labors involved in the projects.
 - Can the resources be scheduled like machines or should they be respected and encouraged?
- Depend on the requirement of (multi-)project strategy, innovation and knowledge management...

Centralization

- Are resources scheduled at a central point, or mainly scheduled by every autonomous project and adjusted when necessary?
- This means different multi-project organization structure and also communication cost across projects or departments.
- Sometimes the degree of centralization or federation can be changed for efficient management, but sometimes hindered by existing mechanism or authority.

Homogeneity

- Projects within a multi-project are usually closely related
 - Managing heterogeneous projects as a multi-project would be much complex
- Have most projects in a multi-project used, or do they suit for one kind of methodology? What is their similarity?
 - common goal, similar size/duration, sharing resources, interlaced scope, potential reusable component, common technology, same business sector, common customer/contractor, compatible methodology...

Complexity & Uncertainty

- Frequently used as indicators when comparing traditional and agile methods
- Complexity
 - Complexity of every project, e.g. team size, mission criticality, team location, domain knowledge gaps
 - Interdependency among projects & project quantity
- Uncertainty
 - Market uncertainty, technical uncertainty and so on

Executive ability

- Different methods ask for diverse executing competence and cost.
 - Established supporting processes or tools, ability and cooperation of participants, may be prerequisite for methods implementation.
 - Organization culture, maturity, complexity and institutionalization also affect methods selection and balancing

Comparison Results

Aspects	Methods										
	<i>Individual</i>	<i>Network</i>	<i>Critical Chain</i>	<i>Resource Modeling</i>	<i>Multi-agent</i>	<i>Virtual Team</i>	<i>Core Team</i>	<i>Pair Programming</i>	<i>Exchange</i>	<i>Multi-product</i>	<i>Classification</i>
Value orientation	●	○	○	●	◐	●	●	●	●	○	◐
Centralization	○	●	●	●	◐	○	○	○	○	●	◐
Homogeneity	○	●	●	◐	◐	◐	◐	○	○	●	◐
Complexity	◐	◐	◐	◐	◐	●	○	○	◐	○	●
Uncertainty	○	○	●	○	◐	◐	●	●	◐	●	●
Executive ability	◐	○	○	○	○	●	●	●	●	●	◐

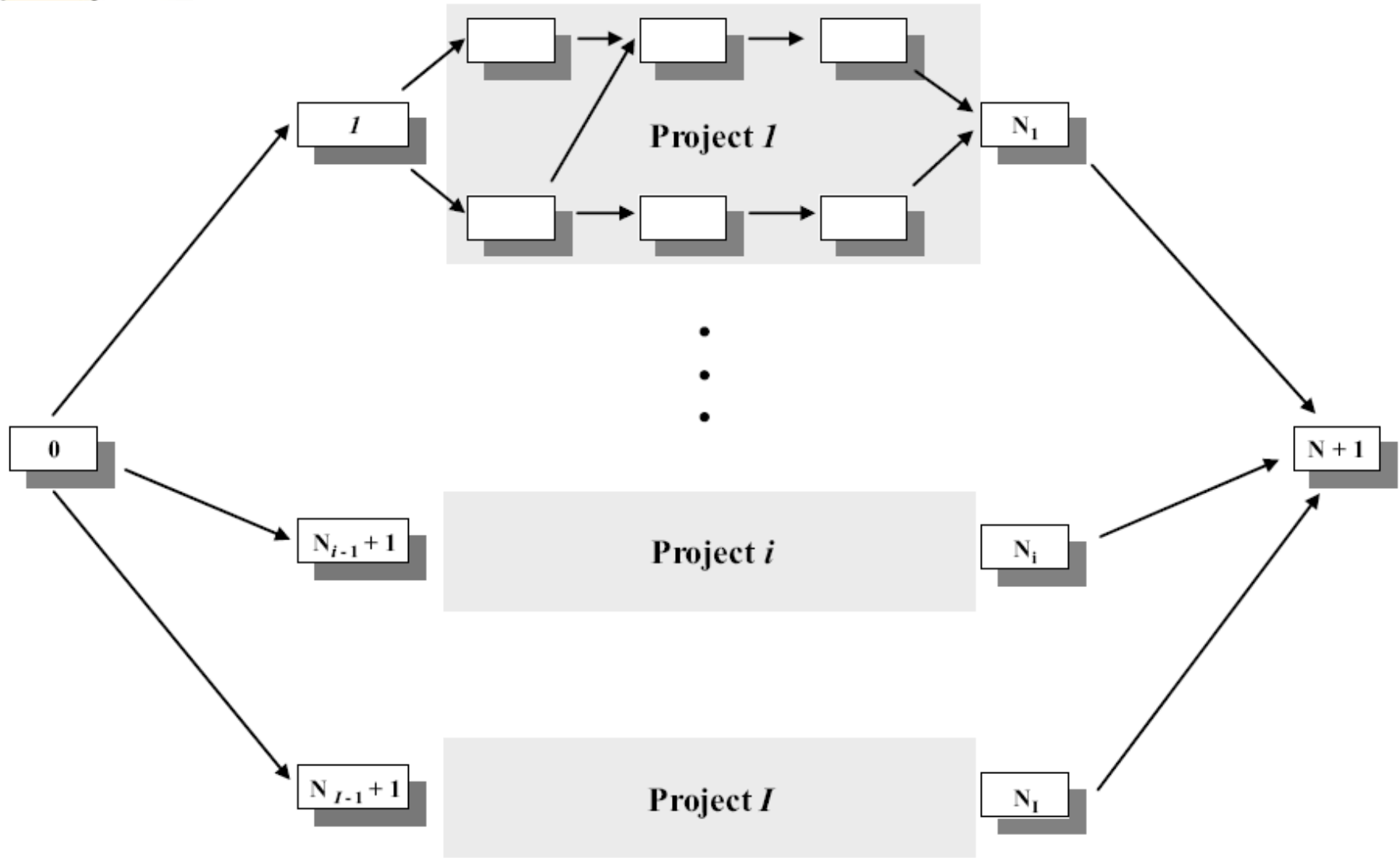
One project at a time

- Manage as individual project/ non-multi-project
- Encourage concentration or creativity rather than resource utilization. **Value orientation**
 - Companies may pursue project quality or lead-time at cost of dedicated and redundant resources.
- Possibly allocate resources centrally before projects start but little interference after carrying out **Centralization**
- Require no similarity between projects **Homogeneity**

One project at a time (cont.)

- Can solve high complexity with managed interfaces
 - It is advised to decompose a mega project into several projects, i.e. a multi-project, to decrease its risk**Complexity**
 - Lose a coping mechanism for uncertainty
 - Ask for resource redundancy for safekeeping**Uncertainty**
 - Ask for enough even redundant resources
 - Depend on document to communicate with related projects
- Executive Ability**

Multi-project as a virtual large project



Networking optimization/heuristics algorithms

- Based on a large multi-project network
- Algorithms dealing with uncertainty in individual PM context can serve as reference
- Various optimization goal or priority rule
 - such as minimum lead time, maximum resource utilization, and minimum project slack
- Treating people as machine

Value orientation

Is multi-project just a large project?

- What is the origin of PM?
- Both cross-functional integration and cross-project management are necessary for effective multi-PM.
- Treating multi-project as a virtual large project ignores the root why a project forms to overcome the difficulties in traditional functional organization.
- General speaking, relevance between projects is weaker than that within a project and project teams should be relatively stable

Networking optimization/heuristics (cont.)

- High centralization **Centralization**
 - allocating resources to “everywhere”
- Multi-project with strongly interrelated projects, e.g. “decomposed” from a common goal, might be scheduled more centrally, like a large project.
- Common supporting tool or exchangeable data format to run (on computer) **Homogeneity**
- Group projects with similar priority together

Networking optimization/heuristics (cont.)

- NP-hard optimization problems **Complexity**
- Some would crash for a large activity network
 - Heuristic procedures are more practical
- Mechanisms to deal with uncertainty in progress **Uncertainty**
- Assume the network of activities known at the start
 - Restructure and optimize again when changes occur
- Computer-supported systems **Executive Ability**
- Integrative system and standard process are required to collect and update data.

Resource modeling & simulation

- Model individuals' knowledge and skills
- Assign right people right task **Value orientation**
- Increase productivity and product quality
- Attach more importance to knowledge&experience
- Basically central methods **Centralization**
- Ask for common supporting tool or exchangeable data format to run (on computer) **Homogeneity**
- Schedule projects sharing no resources
 - by assigning projects corresponding resources

Resource modeling (cont.)

- Model/simulate complex resources or situations **Complexity**
- Even harder to schedule by automatic algorithms
- Model/simulate dynamic availability and capability of resources **Uncertainty**
 - but its scheduling is similar to *Network*
- Computer-supported systems **Executive Ability**
- Integrative system and standard process are required to collect and update data.
- Extra detailed formal description and classification of people's skills and tasks requirements

Fixed core team with flexible resource transfer

- From feature team in Feature-Driven Development
 - Stable and transferable members
- Aim to deal with schedule uncertainty and knowledge sharing or component reuse
- Trust people's initiative and communication
- Schedule autonomously by every project and adjust at a central point such as a meeting
- Common attributes among projects make sense
 - Common technology, potential reusable component...

Value orientation

Centralization

Homogeneity

Fixed core team (cont.)

- From small-and-medium companies **Complexity**
 - Applicability for high complexity remains to be demonstrated.
- Transfer among projects as required to deal with uncertainty **Uncertainty**
- Resource transfer is usually based on negotiation in meeting and relies on people cooperation.
- Synchronize the increment control points of related projects **Executive Ability**

Comparison Results and Analysis

Aspects	Methods										
	<i>Individual</i>	<i>Network</i>	<i>Critical Chain</i>	<i>Resource Modeling</i>	<i>Multi-agent</i>	<i>Virtual Team</i>	<i>Core Team</i>	<i>Pair Programming</i>	<i>Exchange</i>	<i>Multi-product</i>	<i>Classification</i>
Value orientation	●	○	○	●	◐	●	●	●	●	○	◐
Centralization	○	●	●	●	◐	○	○	○	○	●	◐
Homogeneity	○	●	●	◐	◐	◐	◐	○	○	●	◐
Complexity	◐	◐	◐	◐	◐	●	○	○	◐	○	●
Uncertainty	○	○	●	○	◐	◐	●	●	◐	●	●
Executive ability	◐	○	○	○	○	●	●	●	●	●	◐

Discussion (cont.)

- Some methods emphasize on people while others on process: classical balance in software community
- Methods can be roughly classified into 3 categories: heavy-weight, light-weight, and medium
- Some similarity among projects is required
- Autonomic management with cooperation is more efficient than central management and individual PM

Conclusion and Future Work

- An initial attempt to compare and evaluate the resource scheduling methods
- Which of the methods are more useful and suitable to different application environments and constraints
- A more elaborate matching algorithm with measurable factors that actively contributes to the aspects
- Case study and questionnaire
- Multi-project situation pattern
- A consolidated method with more scalability

Thank You
Question & suggestion?