### Outline
- What’s the current problem for large software projects
- What’s the previous solution for large projects
- What the paper presents
  - Dataset
  - InfoVis Encoding Technique
  - Visualization Method
  - Sample Usage
  - Future Improvement

### Previous work – Sankey Diagram
Sankey diagrams are a specific type of flow diagram, in which the width of the arrows is shown proportionally to the flow quantity.

**Problem:** No Time Frame

### Previous work – Gantt Chart
A Gantt chart is a type of bar chart that illustrates a project schedule. Gantt charts illustrate the start and finish dates of the involved tasks. Modern Gantt charts also show the dependency.

**Problem:**
1. Doesn’t show how many people/resources involved in project
2. Don’t have a whole picture about the project

### What’s the author propose
- A graph flow which can not only show the interconnection of different modules of development along with the timezone but also the programmer who took part in the whole project with vivid color to show difference

### Dataset: Developer Activity Model
1. Abstract commit as $c$, time as $t$, developer name as $d$, files as $f$, all files as $F$, file modules hierarchy as $H$

- Partition sequence of commits into equally sized intervals for each interval of commits and many modules
- Calculate individual developer activity of files for each module and each developer
- Calculate weighted activity matrix for each developer $A = MAK * N^{(1+1)}$
- Calculate weighted activity matrix for all developers by scoring up $A$

**Paper didn’t show how to map real data into Activity Model Matrix**

### Visualization Technique
- Dataset: Developer Activity Model
- Encoding: Develop River for Time-Varying Developer Activities

### Developer Rivers Curves
1. Transition: how developers change their behavior between different module groups using cubic Bezier Curves
2. Transition color as a linear gradient from color of start module to target module

**Paper didn’t show how to link Matrix Data with Bezier curve creation**

---

**Visualizing Work Process in Software Engineering with Developer Rivers**

**Presenter:** Arthur Sun

Michael Burch, Tanja Munz, Fabian Beck, and Daniel Weiskopf

VISUS, University of Stuttgart, Germany

**Large open-source projects: 441 releases**

**GitHub**

**Large open-source projects: 560,519 commits**

**GitHub**

---

**Visualizing Work Process in Software Engineering with Developer Rivers**

**Presenter:** Arthur Sun

Michael Burch, Tanja Munz, Fabian Beck, and Daniel Weiskopf

VISUS, University of Stuttgart, Germany

**Previous work – Sankey Diagram**

Sankey diagrams are a specific type of flow diagram, in which the width of the arrows is shown proportionally to the flow quantity.

**Problem:** No Time Frame

**Previous work – Gantt Chart**

A Gantt chart is a type of bar chart that illustrates a project schedule. Gantt charts illustrate the start and finish dates of the involved tasks. Modern Gantt charts also show the dependency.

**Problem:**
1. Doesn’t show how many people/resources involved in project
2. Don’t have a whole picture about the project

**What’s the author propose**

- A graph flow which can not only show the interconnection of different modules of development along with the timeframe but also the programmer who took part in the whole project with vivid color to show difference

---

**Dataset: Developer Activity Model**

1. Abstract commit as $c$, time as $t$, developer name as $d$, files as $f$, all files as $F$, file modules hierarchy as $H$

- Partition sequence of commits into equally sized intervals for each interval of commits and many modules
- Calculate individual developer activity of files for each module and each developer
- Calculate weighted activity matrix for each developer $A = MAK * N^{(1+1)}$
- Calculate weighted activity matrix for all developers by scoring up $A$

**Paper didn’t show how to map real data into Activity Model Matrix**

---

**Visualization Technique**

- Dataset: Developer Activity Model
- Encoding: Develop River for Time-Varying Developer Activities

---

**Developer Rivers Curves**

1. Transition: how developers change their behavior between different module groups using cubic Bezier Curves
2. Transition color as a linear gradient from color of start module to target module

**Paper didn’t show how to link Matrix Data with Bezier curve creation**

---

**Large open-source projects: 5659 contributors**

**GitHub**

---

**What’s do we want**

A whole picture of the overall progress of extreme large software engineering project proceeding with time frame in detailed visualization for major participants, their contribution to respective work, how much amount of work they did and their work change

---

**Large open-source projects: 441 releases**

**GitHub**

---

**So many changes in Commit, Releases, Contributors!**

How can I see it in a whole picture?**

---

**Previous work – Gantt Chart**

A Gantt chart is a type of bar chart that illustrates a project schedule. Gantt charts illustrate the start and finish dates of the involved tasks. Modern Gantt charts also show the dependency.

**Problem:**
1. Doesn’t show how many people/resources involved in project
2. Don’t have a whole picture about the project

---

**Previous work – Sankey Diagram**

Sankey diagrams are a specific type of flow diagram, in which the width of the arrows is shown proportionally to the flow quantity.

**Problem:** No Time Frame
One of the most obvious observations in Figure 1, which shows a qualitative change of developer activity (i.e., developers’ relative efforts switch back and forth between the different modules. Hence, the library code seems to have gained importance in relation to C code.

Fig. 5. Python Developer Sparkline of top 5 developers

Fig. 6. Python Developer Sparkline of top 5 developers

Fig. 7. Python Subsystem details of Tools Directory

Fig. 8. Linux Subsystem details of Tools Directory

Diagram

- Inflow/Outflow: A transition from or to the outside of the diagram identifies a developer entering or leaving the project.
- Constant Flow: An in- or out-transition with a constant width indicating a group of developers constantly working on the same module.
- Growth/Decline: An in- or out-transition with an increasing or decreasing strength hinting at a group of developers that keep working on a module but with changing total effort.
- Split/Merge: A module that is split into or merged from multiple flows shows a qualitative change of developer activity (i.e., developers’ relative focus switches between modules). While at least one inter-transition is required for this pattern, one of the flows can be an intra-transition.
- Exchange: A pair of intra-transitions connecting two modules in opposite directions at the same time is a specific qualitative change of activity: some developers move between the two modules in both directions.

Visual Patterns

- Main Module Overview
  - Connects main directories, developers and their contributors
- File Type Overview
  - Automatic definition of modules by file types
- Developer Sparklines
  - Highlight top 5 star developer contributing most to the whole project
- Subsystem Details
  - Modules in a subdirectory of the system show details of a specific system

Future Improvements

- Show us how do the author organize the data (Data >Matrix)
- Show how to transfer the data into influents and effluents (Matrix >Influents)
- Provide tool ready for practitioners who can use developer river directly (no description about how to tackle the dataset)
- Distinguished colors may be up to 10 colors, otherwise graph may be hard to see
- Transfer the way to study software engineering research into social-technical aspects of engineering research.