# Xsemble

Visual Assembly of Software

***Make software like they make cars!***



September 2020

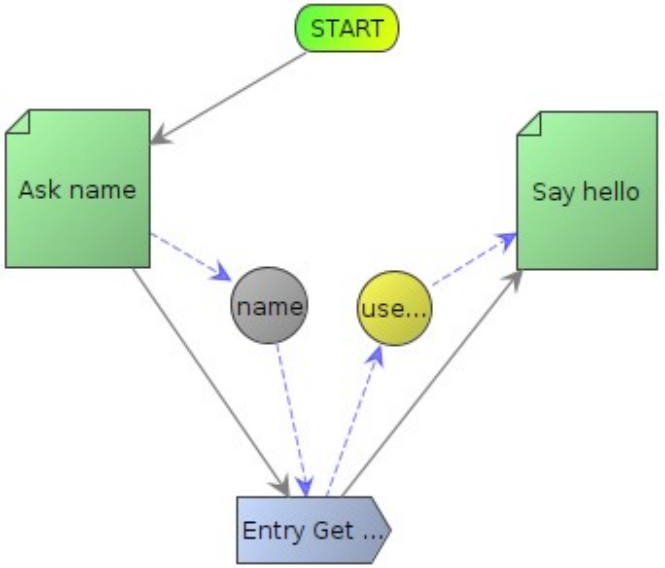
# Introduction

**There are many common challenges faced in the development of software.**

- Business analysts are often blind to the design of how software works.
- Onboarding a developer in existing software project is tough because of the steep learning curve.
- Enhancements become difficult as software grows in complexity.

**The new Xsemble technology offers a solution to these challenges.**

# What is Xsemble?



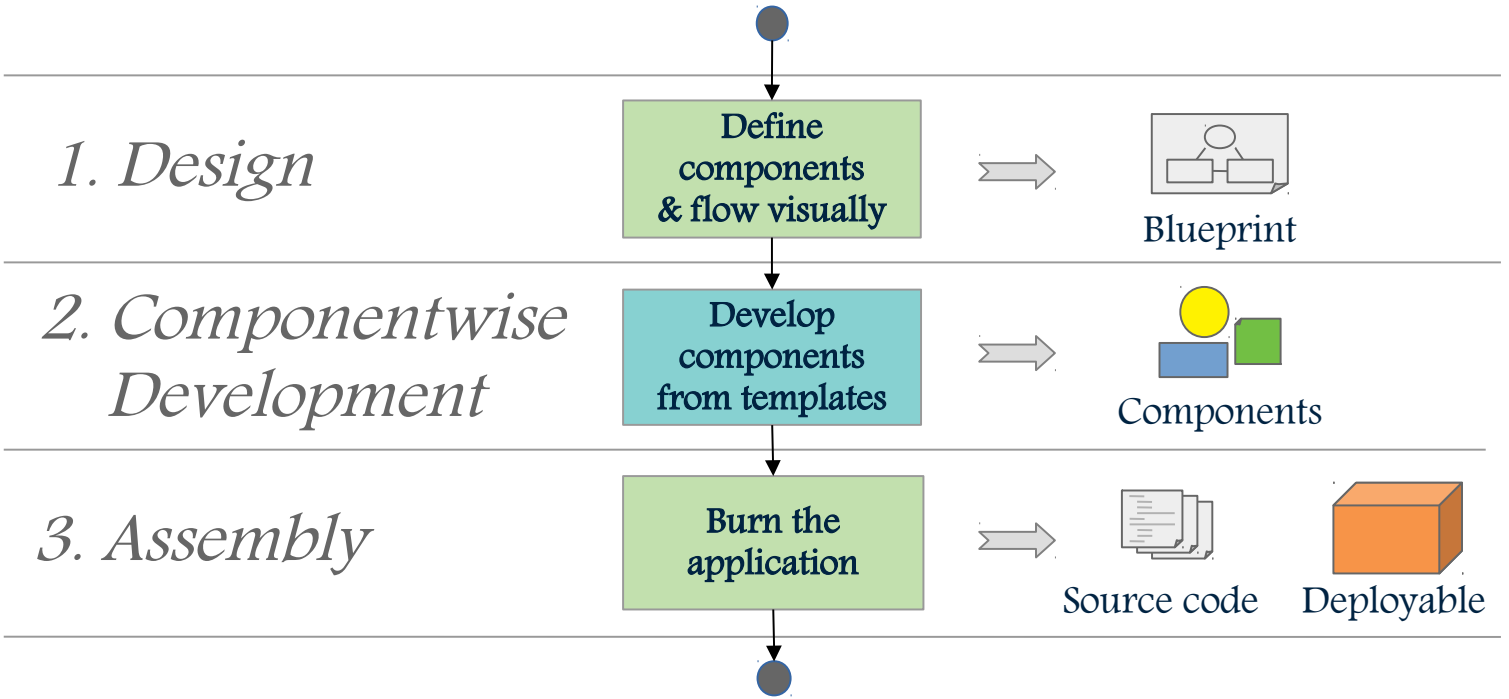

- Xsemble is a low-code technology solution (patent filed) for software development.
- It enables “Visual Assembly of Software”.

# Why Choose Xsemble?

| Benefits For | Key Benefits |
|---|---|
| **Software product companies** | ✔ Saving effort in development (~40%) and maintenance (~80%) |
| **Software services companies** | ✔ Flexible teams<br>✔ Efficient utilization of IT and non-IT resources<br>✔ Crisp project management<br>✔ Evolving repositories of reusable components |
| **Non-IT companies** | ✔ Creating IT assets while reducing vendor dependence<br>✔ Better control within the organization |
| **Startups** | ✔ Saving costs<br>✔ Reducing technology risk<br>✔ Quicker time to market |
| **Educational institutes** | ✔ Educating students on software design and development |

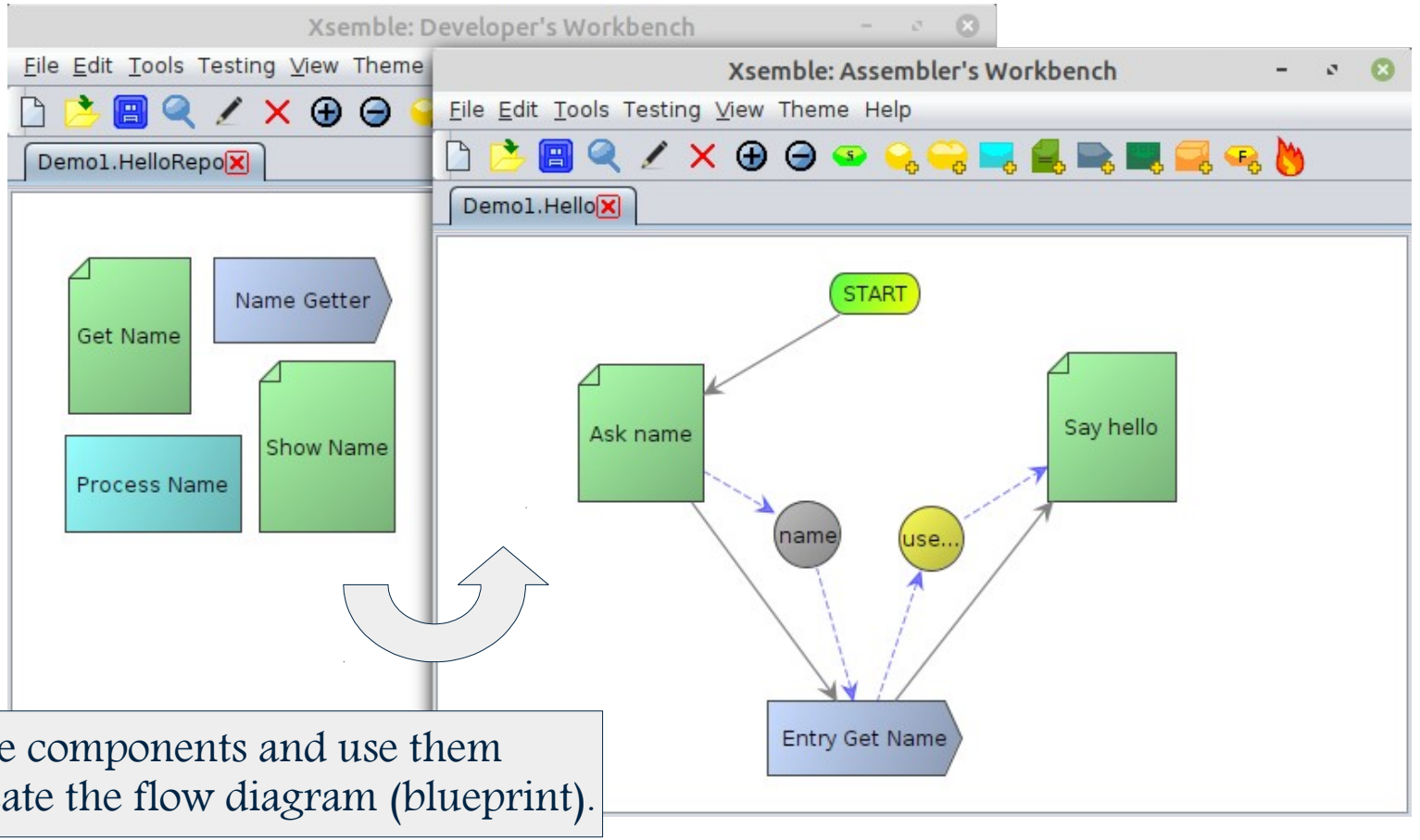
# Xsemble Process

**3 step process. Like cars are made. Executed in waterfall or agile mode.**




**Programming is not required for steps 1 & 3.**

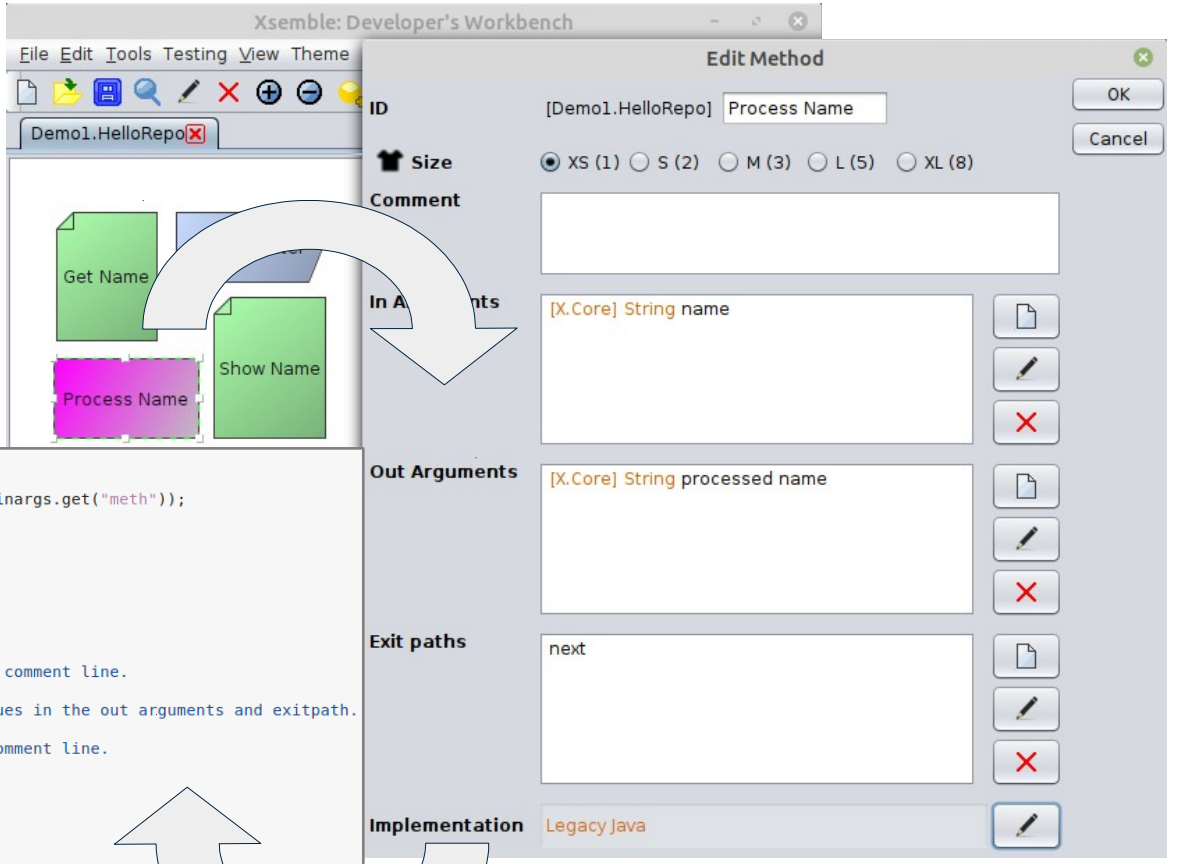
# Step 1 – Design




Create components and use them
to create the flow diagram (blueprint).

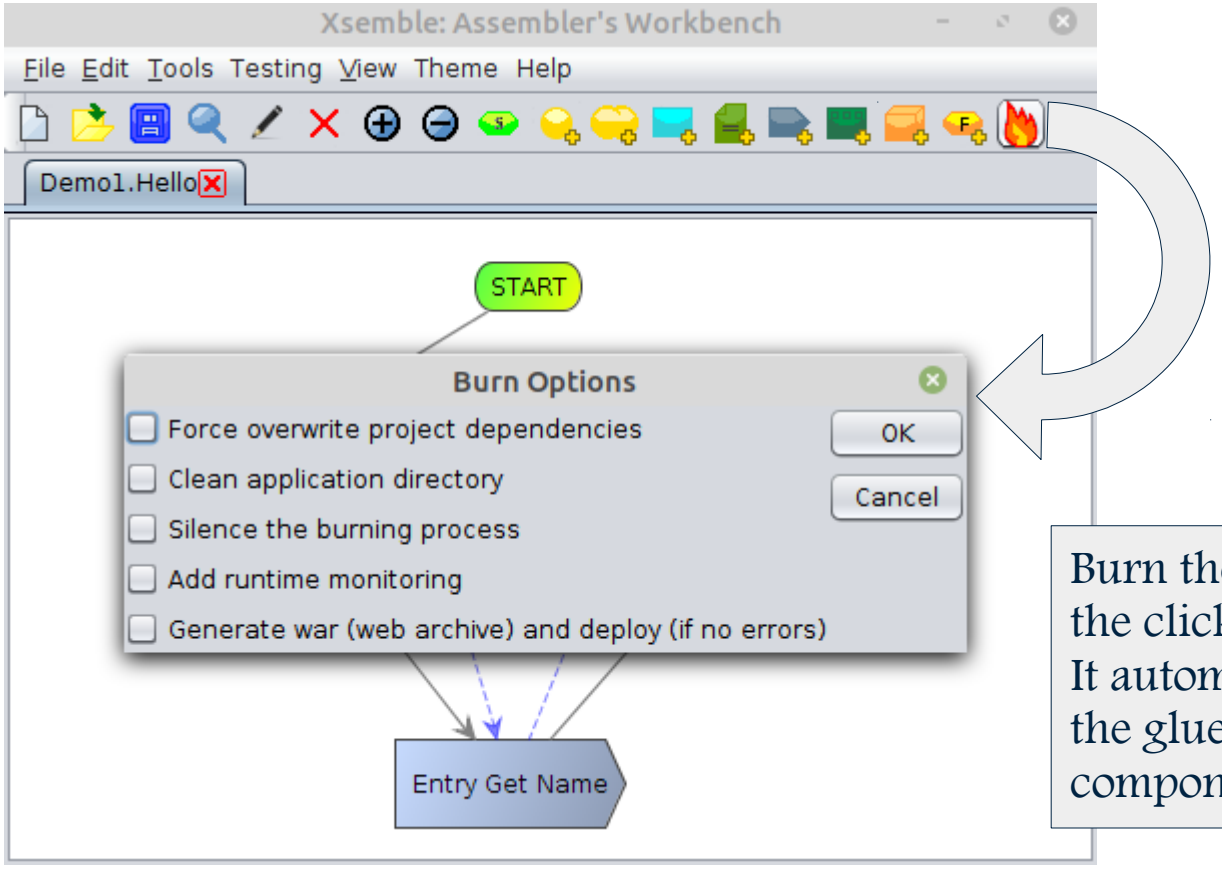
# Step 2 – Componentwise Development

Generate implementation code template.
Let programmers add the desired functionality.



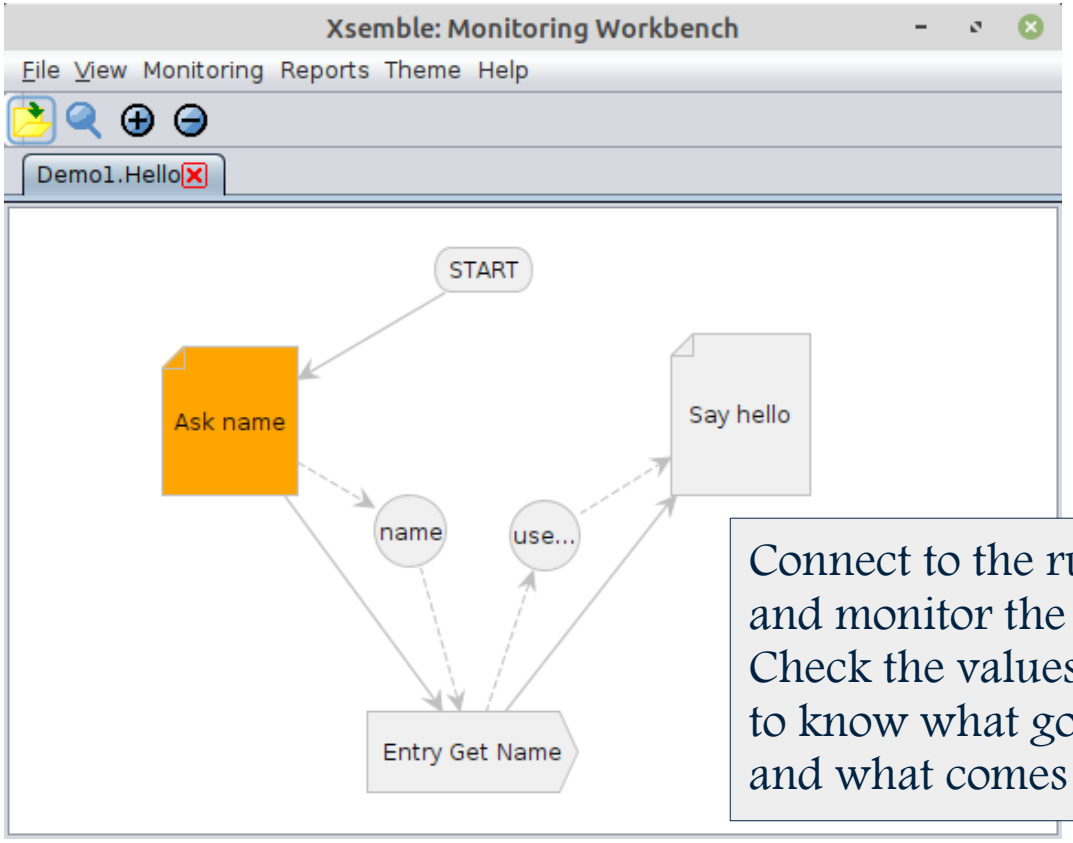

```
31     public Map<String,Object> run(Map<String,Object> inargs)
32     {
33         logger.info("Session:"+ inargs.get("session")+" - Entered method " + inargs.get("meth"));
34         // Get the in arguments
35         java.lang.String name_in = (java.lang.String) inargs.get("name");
36
37         // Out argument declarations
38         java.lang.String processed_name_out;
39
40         // Method exit path
41         String exitpath = "next";  // Possible values are "next"
42 // USER-CODE-BLOCK BEGIN -- Do not modify anything outside this block or this comment line.
43
44         // TODO: Your implementation comes here. Populate the appropriate values in the out arguments and exitpath.
45
46 // USER-CODE-BLOCK END -- Do not modify anything outside this block or this comment line.
47
48         // Create outargs and populate it
49         HashMap<String, Object> outData = new HashMap<String, Object>();
50         outData.put("_exitPath",exitpath);
51         outData.put("processed name", processed_name_out);
52
53         return outData;
54     }
```

# Step 3 – Assembly



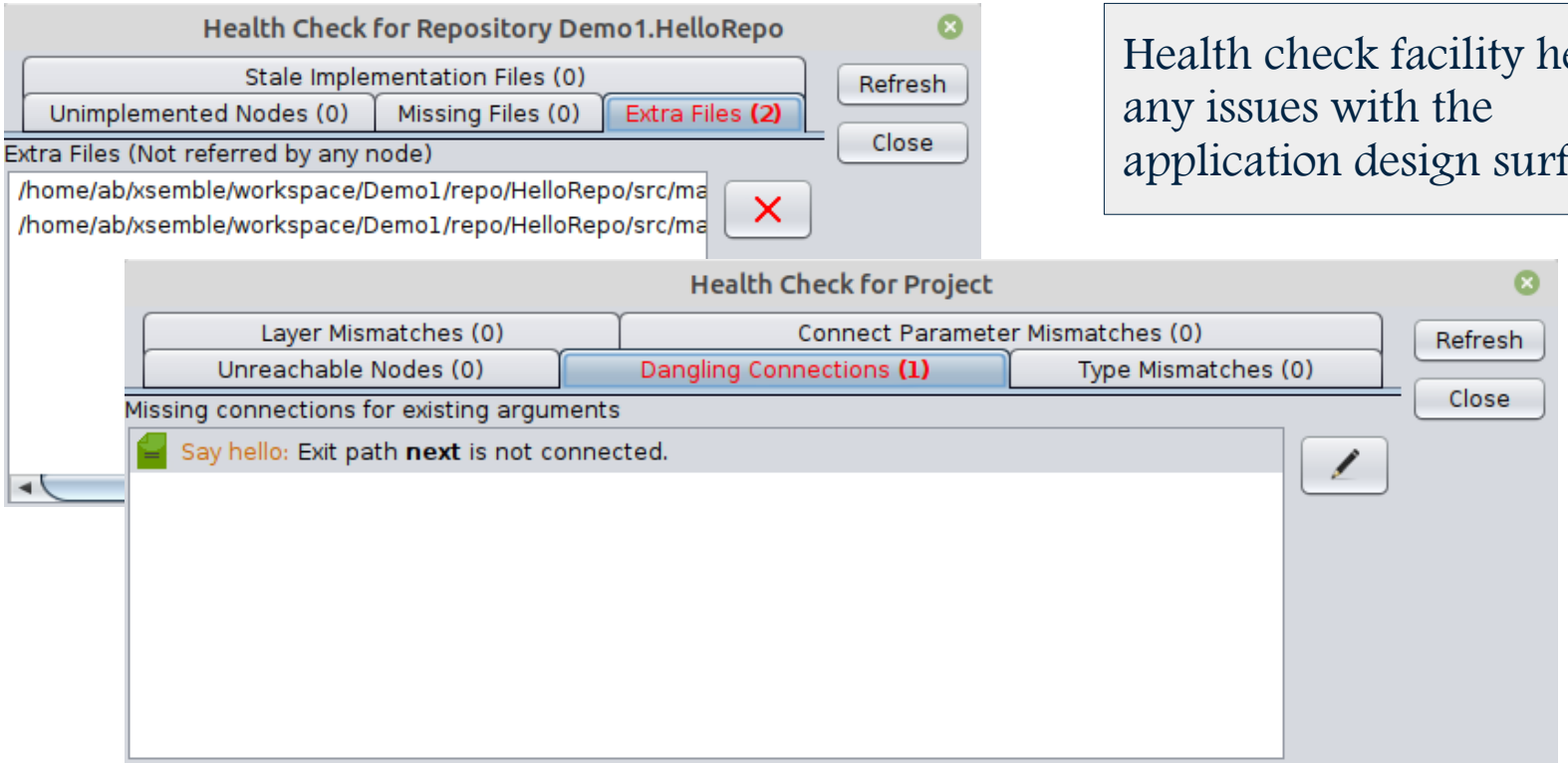

Burn the application at the click of a button.
It automatically generates the glue code to wire the component implementations.

# Runtime Monitoring



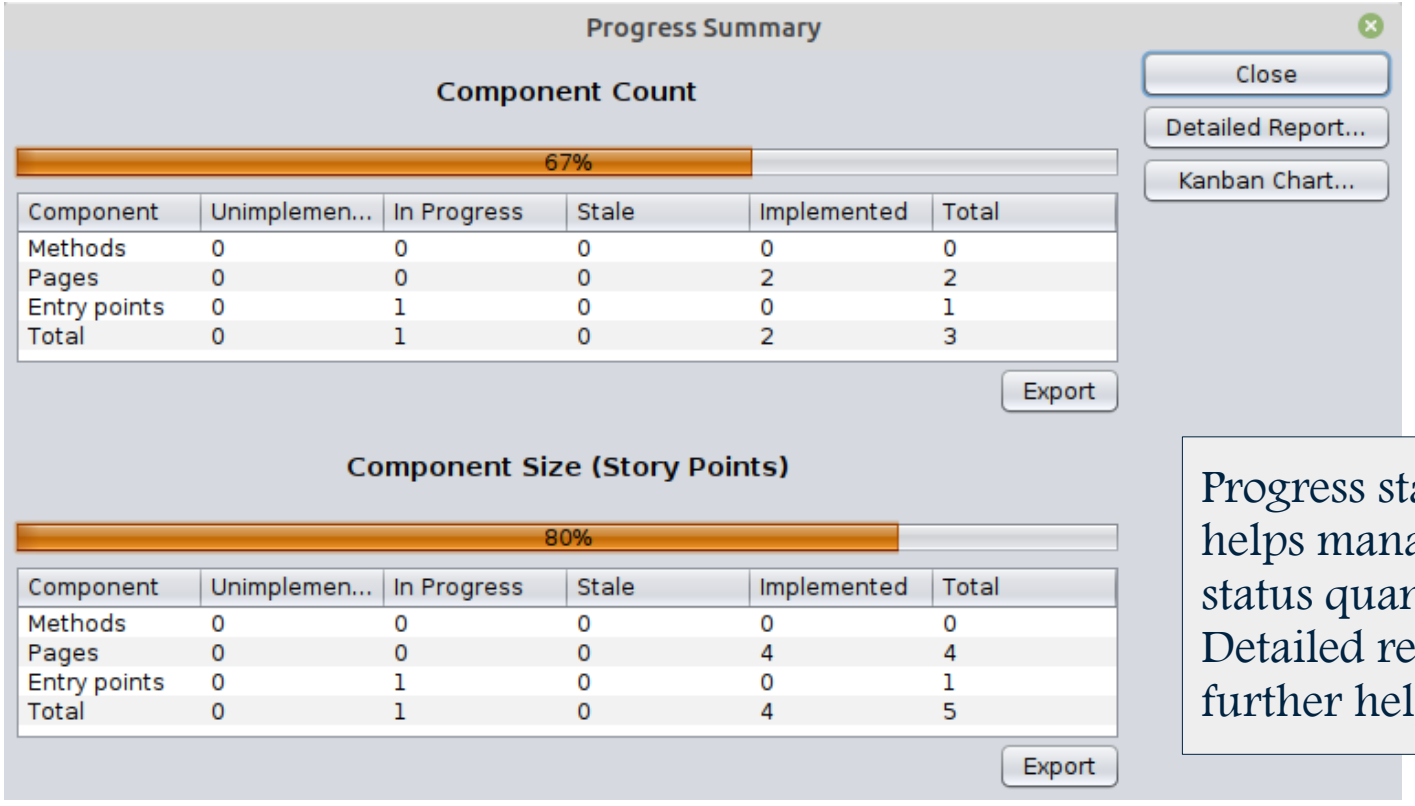

Connect to the running application and monitor the flow visually. Check the values of the variables/ to know what goes into a node and what comes out of it.

# Automatic Check of Design




Health check facility helps any issues with the application design surface.

# Aiding the Managers




Progress statistics at the fingertips helps managers know the exact status quantitatively.
Detailed report and Kanban chart further help in drilling down.

# Impact – Change in Roles

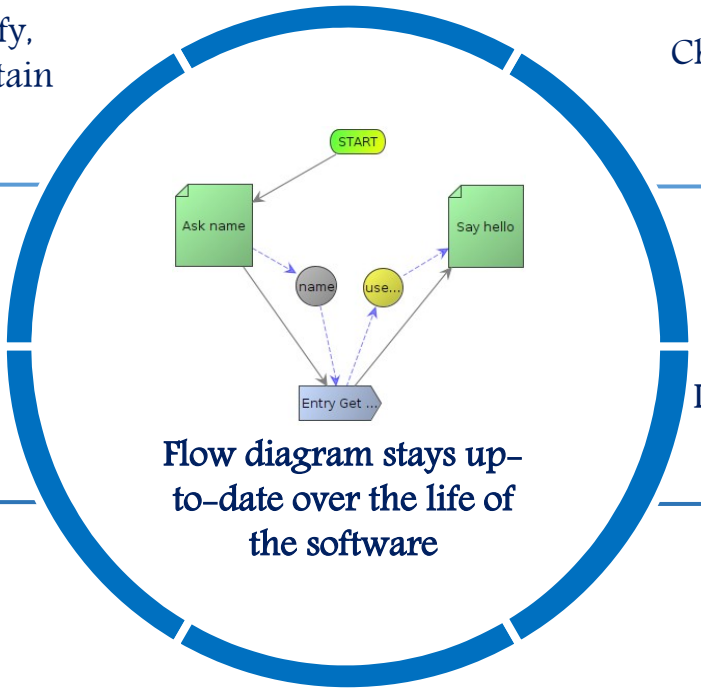

| End User | Assembler | Programmer |
|---|---|---|
| • Role **does not change**.<br>• Gets the application quicker, with less errors<br>• Their issues are addressed quickly and satisfactorily | • **No programming needed**.<br>• Domain experts or Managers can take it up.<br>• Saves the effort of teaching domain nitty-gritties. | • Programming becomes **easier**, from coding entire application to coding component implementations.<br>• They get to **focus on their core strength – programming**. |

# Unprecedented Visibility

**Domain Expert** — Review, Certify, Create & maintain

**Manager** — Plan work, Track progress, Assess risk

**Other Stakeholders** — Review, Understand




**Flow diagram stays up-to-date over the life of the software**

**Programmer UI/UX** — Onboarding, Choose components, Unit testing

**Quality Engineer** — Test planning, Troubleshoot, Defect assignment

**Support Engineer** — Troubleshoot, Forward issue

# Development Phase Effort Saving

Waterfall or Agile, SDLC (Software Development Life Cycle) captures the phases in software development.

| SDLC Phase | Conventional | With Xsemble | Reasons for Reduction |
|---|---|---|---|
| ➢ Requirements | 15 | 10 | ✓ Technical resources not needed |
| ➢ Design | 25 | 20 | ✓ Health Check helps in design validation<br>✓ Reliable, granular estimate from component sizing |
| ➢ Development | 40 | 20 | ✓ Generated code and unit test templates, Generated glue code<br>✓ Smaller components ► Accuracy ► Reduced rework<br>✓ Better progress tracking at component level |
| ➢ Testing and Defect fixing | 15 | 7 | ✓ Higher visibility, more accuracy ► Less defects<br>✓ Ease of fixing bugs as they can be traced to component level |
| ➢ Deployment and User acceptance | 5 | 3 | ✓ Visibility into flow diagram ► Omissions less likely<br>✓ Changes at component level ► Easy to make |
| Total | 100 | 60 | |

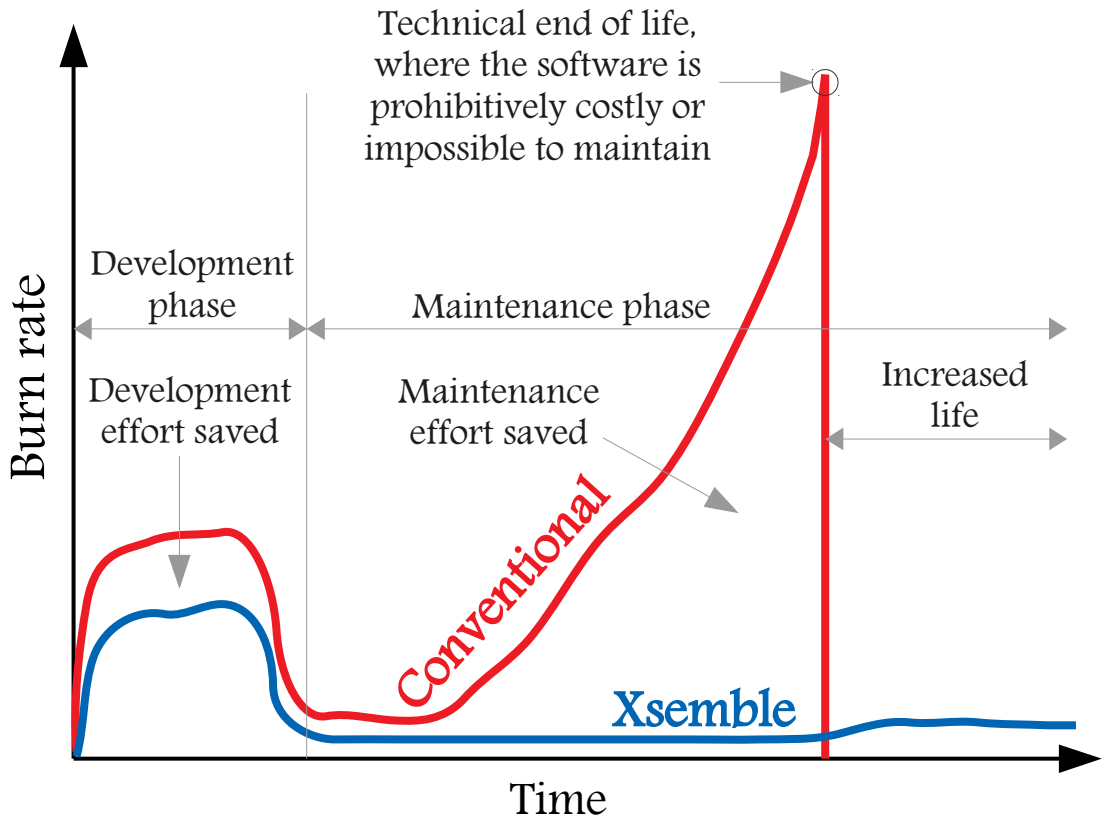
Note: Actual savings vary based on engagement-specific situations.

# Maintenance Phase Effort Saving

Conventionally, maintenance is 85% of the total software cost.
Using Xsemble, 80% savings in maintenance cost is estimated.

| Conventionally | With Xsemble |
| --- | --- |
| • Effort keeps **increasing expontially**, as team members change and code becomes patchy.<br>• At some point, software cannot be maintained anymore. A **re-engineering is forced**.<br>• Team members having **high familiarity with existing codebase** is very important. | • Using **visual monitoring**, an error can be traced to a component *before* involving a programmer.<br>• Programmer does a ***surgical* fix**. This does not need familiarity with the entire system. **Fast and accurate**.<br>• **New team members** can be onboarded easily, as component level coding is orders of magnitude little to understand. |

# Summary of Effort Saving




Effort Saving = Area between the red and the blue curves

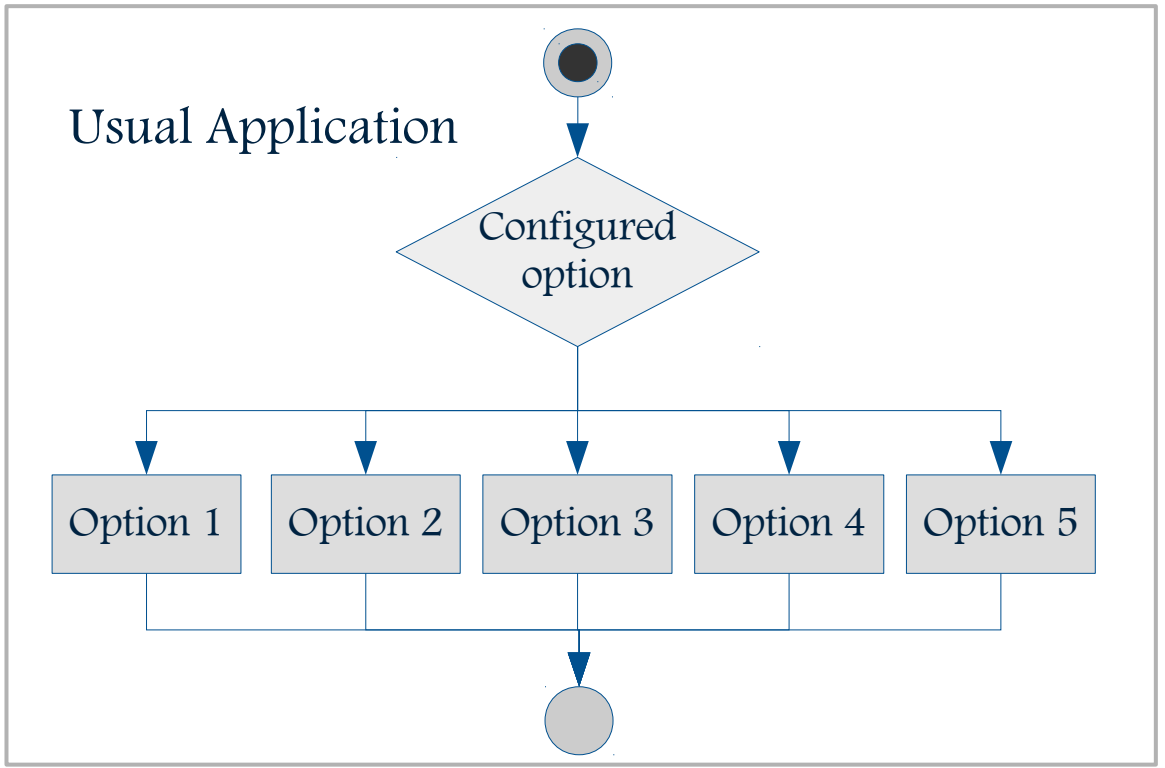
Possible to switch to the blue curve from the red curve

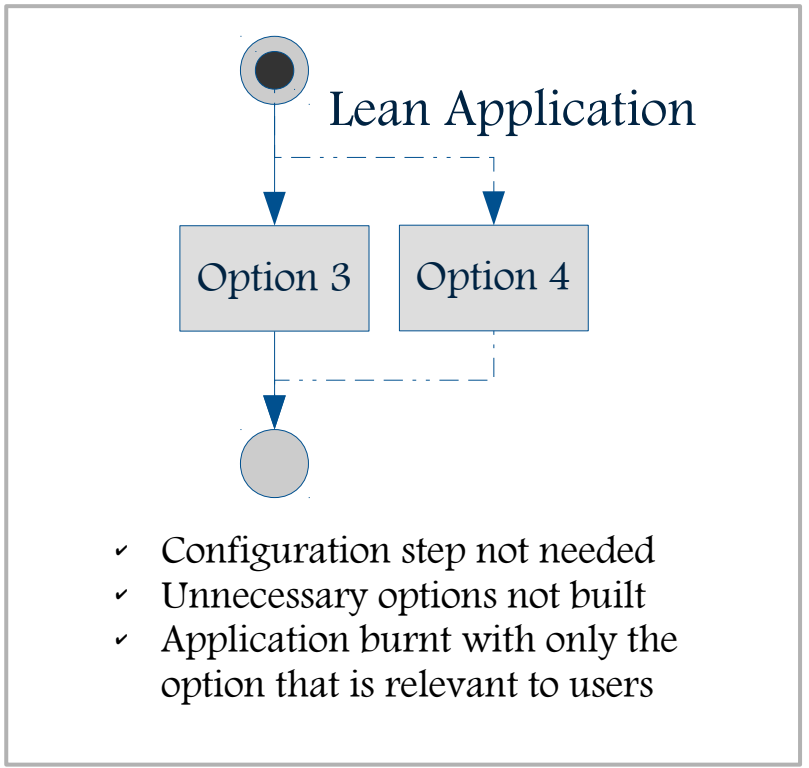
- One time Re-engineering effort
- Rewarded with further effort saving
- Sooner the better!

# Application Optimization

**Build what you need, when you need.**
**Lean applications take less to build and perform better.**

Click for further reading

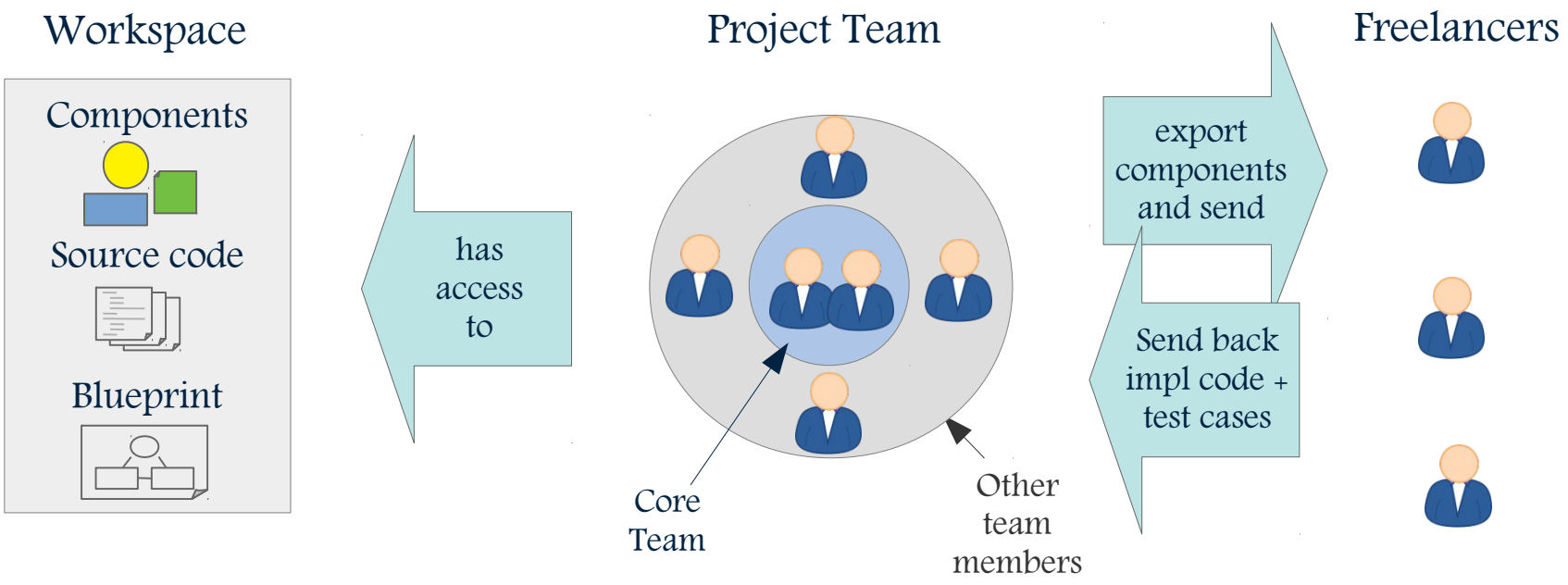



- Configuration step not needed
- Unnecessary options not built
- Application burnt with only the option that is relevant to users

# Taking Outside Help

**Using test bundle mechanism, you can take external help for select components.**



# Financial Impact

| Business Opportunities | Cost Savings |
| --- | --- |
| • Lower Time-to-Market<br>• Better Quality<br>• Longer shelf life | • Effort saving<br>• Smart resourcing<br>• DIY (Do It Yourself)<br>• Easier maintenance |

# Frequently Asked Questions

- **How is the code shared among developers?**
  Conventionally, developers use repositories such as git or svn for versioning and sharing the code with one another. The same mechanism can be used with Xsemble workspace. It contains all the Xsemble artefacts.

- **Is it possible to stop using Xsemble at some point?**
  Yes. You have access to the complete source code of the resultant application, and you can take it and run with it without Xsemble at any time. (But we hope you do not do that.)

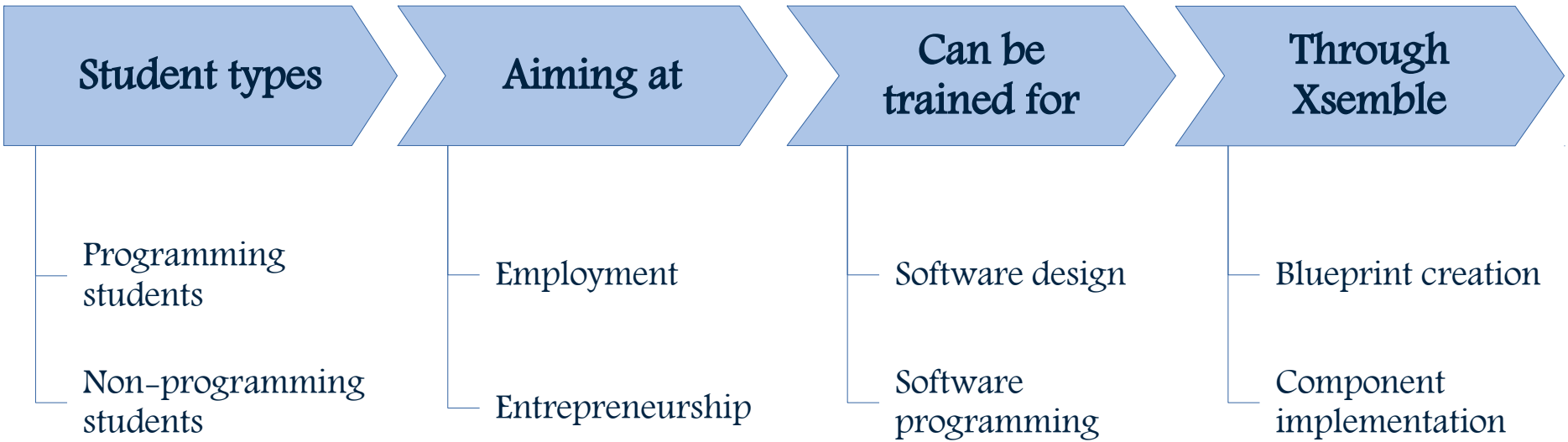
- **Is the IP of component implementations secured?**
  Your component implementation code is never sent to Xsemble cloud and stays on your machine. Its security on your machine is completely within your control.

# Market Validation – Case Studies

Xsemble is used in creating applications for various domains.

| | |
|---|---|
| **IoT** | 2 small applications prepared & delivered by 2 freshers in <15 days |
| **HR** | Built a competency matrix creation tool using Natural Language Processing. Showcasing Xsemble's ability to utilize any technology, including deep tech. |
| **Accounting** | Amazon MWS API integration for fetching billing data, to import it into an accounting software. |
| **Legal Process Outsourcing** | The customer purchased Xsemble licenses and drove the development effort of this complex application. |
| **Recruitment** | Complex application "HeadTracker" built by Acism, consisting of 1231 nodes. |
| **Supply Chain** | Custom reports primarily meant for supply chain and inventory management. |
| **Industry Chamber** | A custom content management system developed for an association of industries. |

# Xsemble for IT Education




**Ashish Belagali**
ab@xsemble.com
+91 98900 56365
https://xsemble.com
Twitter @Xsemble