

Apache Lenya: Roadmap

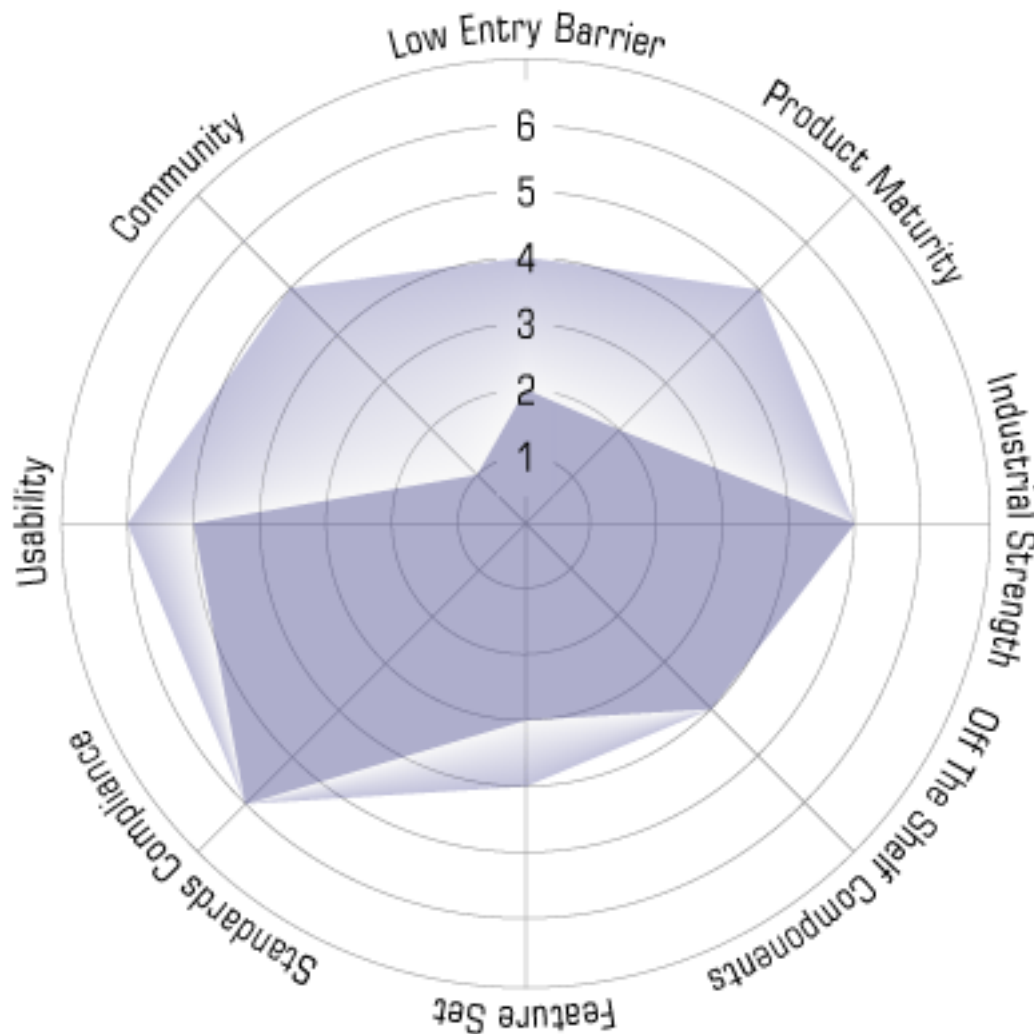
Table of contents

1 Positioning Diagram.....	3
2 Releases.....	4
3 Long-term Goals.....	4
4 Release 1.0.....	4
4.1 Community.....	4
4.2 Low Entry Barrier.....	4
4.3 Product Maturity.....	5
4.4 Industrial Strength.....	5
4.5 Off The Shelf Components.....	5
4.6 Feature Set.....	5
4.7 Standards Compliance.....	5
4.8 Usability.....	5
5 Release 1.2.....	5
5.1 Community.....	5
5.2 Low Entry Barrier.....	5
5.3 Product Maturity.....	6
5.4 Industrial Strength.....	6
5.5 Off The Shelf Components.....	6
5.6 Feature Set.....	6
5.7 Standards Compliance.....	6
5.8 Usability.....	6
6 Release 2.0.....	6
6.1 Community.....	6
6.2 Low Entry Barrier.....	7
6.3 Product Maturity.....	7
6.4 Industrial Strength.....	7
6.5 Off The Shelf Components.....	7
6.6 Feature Set.....	7

6.7 Standards Compliance.....	7
6.8 Usability.....	7

1. Positioning Diagram

The Lenya roadmap is determined by the Lenya positioning diagram. It specifies the priorities and orientation, and shows areas for further development. The 1.0 release marks the point where Lenya was donated to the Apache foundation. Subsequent releases have been under the Apache umbrella. Items in bold type mark completed items.



The positioning diagram defines 8 areas, with a relative weight of 1-7. The current Lenya code base is represented by the dark blue area, while the light blue area indicates focus areas for future versions. The positioning diagram is based on a hard look at where we currently stand. The various areas are defined below:

1. Community

How viable is the community supporting the product? Are there independent contributors? How active are mailing lists, web sites? Is the community healthy, friendly and welcoming to newcomers?

2. Low Entry Barrier

How easy is it to get started? Is the product a turn-key solution with immediate value after download? How consistent is the documentation, how hard is it to familiarize oneself with the codebase?

3. **Product Maturity**

How robust is the underlying code? Are sound software engineering practices in use? Is the architecture clean and consistent? Are changes to the code regression-tested?

4. **Industrial Strength**

How much does the product scale? How flexible and stable is the product, how well does it perform? Is it secure?

5. **Off The Shelf Components**

How much does the product make use of standard components? Is the architecture modular to accomodate external code?

6. **Feature Set**

Does the product have a conservative feature set, or does it have many bells and whistles?

7. **Standards Compliance**

How much does the product use and interoperate with standards?

8. **Usability**

Is the user interface understandable for non-technical persons? Is it consistent? Does it follow usability best practice?

2. Releases

Three releases are planned: 1.0, 1.2 and 1.4 (subsequently called 2.0), with the possibility of smaller maintenance releases in between. Even numbers are stable releases, odd numbers development releases. Minor releases are used for bug fixes, e.g. 1.2.2.

- Release 1.0 Summer 2003
- Release 1.2 Fall 2003
- Release 2.0 - coming soon

3. Long-term Goals

- Improve communication and collaboration with sister Apache projects

4. Release 1.0

4.1. Community

- **Migrate mailing lists / CVS to Apache**
- Provide scratchpad for third party publications

4.2. Low Entry Barrier

- Provide Class Diagrams
- **Disentangle CMS and publications**
- **Create outline of new documentation**

- **Provide default publication**

4.3. Product Maturity

- **Refactor code base**
- **Remove dead code**
- **Provide unit tests for core classes**
- **Simplify directory structure**
- **Provide default implementations in root sitemap, with overriding capability**

4.4. Industrial Strength

- **Update to current Cocoon 2.1 snapshot**
- **Load test and provide benchmarks**
- **Add Unit test framework**
- **Disentangle CMS and Cocoon**

4.5. Off The Shelf Components

- **Update from JDK 1.3.1 to 1.4**
- **Replace DOM4J**

4.6. Feature Set

- **Make windows version usable again**
- **Finish Lucene integration**
- **Bugfree Revision Controller**

4.7. Standards Compliance

- **Enforce Avalon coding standards**

4.8. Usability

- **Standardize CMS screens**
- **Standardize Lenya menus**

5. Release 1.2

5.1. Community

- **Get additional committers on board**
- **Define scope and collaboration with the Forrest project**

5.2. Low Entry Barrier

- **Improve and standardize templating (XHTML)**
- **Allow for documentation annotation (like php.net)**

- **Write a publication HOWTO**
- **Maintain a FAQ**
- **Eliminate redundancies between authoring and live areas**
- **Write Lenya Tutorials**

5.3. Product Maturity

- **omplete JavaDoc**
- **Unify CMS URIs**
- **Standardize sitemaps**
- **Work out upgrade schedule for dependent components (Cocoon etc)**

5.4. Industrial Strength

- **Substantial Unit test coverage**
- **Add web tests**

5.5. Off The Shelf Components

- **Migrate Lenya xinclude processor to Cocoon (or vice versa)**

5.6. Feature Set

- **Integrate Repository**
- **Implement workflow engine**
- **Implement generic sitetree navigation framework**
- **Integrate HTML Form Editor**
- **Provide GUI for user administration**

5.7. Standards Compliance

- **Provide LDAP authentication**

5.8. Usability

- **Improve Lenya menu to float across the page**
- **Improve CMS screens with help tooltips**
- **Improve Bitflux / Xopus integration**

6. Release 2.0

6.1. Community

- **Work with OSCOM.org on CMS standards**
- **Organize Lenya Sprints**
- **Leverage other Apache projects**

6.2. Low Entry Barrier

- Improve documentation

6.3. Product Maturity

- **Rewrite Resource Creation Framework**
- **Rewrite Publishing / Replication Framework**
- Complete Unit test coverage
- **Complete JavaDoc**

6.4. Industrial Strength

- **Improve reverse proxying**
- **Research advanced caching strategies**
- **Implement transaction management**

6.5. Off The Shelf Components

- Consolidate Authentication and Authorization components

6.6. Feature Set

- Provide (Open) Office integration
- **Implement link management**
- **Implement multilingual capabilities**

6.7. Standards Compliance

- Refine and review standards support

6.8. Usability

- **Provide administration GUI**
- Complete the move to an in-place UI paradigm