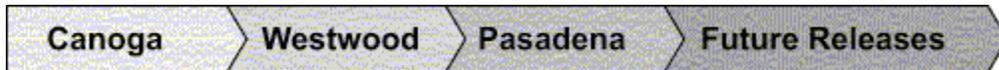




OPEN SOURCE APPLICATIONS FOUNDATION

Chandler - Product Roadmap



Executive Summary

We anticipate the Chandler product to undergo four distinct phases of development:

We anticipate the Chandler product to undergo four distinct phases of development:

1. In the Canoga Early Developer Releases (Chandler 0.2-0.5 release), we will focus on platform and infrastructure, designing and testing out our compelling innovative capabilities, putting in place a modular architecture and providing basic elements of end-user applications to validate our platform.
2. In the Canoga End User Releases (Chandler 0.5-1.0), we will target the info-centric early adoptor by building strong info-management capabilities, power email features, basic calendar and contacts functionality with sharing and collaboration as an integral piece of the product.
3. In the Westwood Releases, we will target higher education users by providing rich, standards-based calendar capabilities, interoperability with existing infrastructure, central server support for nomadic access, and other university-specific requirements.

The Westwood release will be the first release we believe will be satisfactory for deployment within universities. It will have excellent workgroup support but only rudimentary central server support and maturity. Central server features will not be ready for campus-wide usage but will be ready for pilot test and smaller workgroup deployments.

After Westwood, we will work on issues and functionality such as server robustness, stronger and more specialized administration tools, a CAP-compliant calendar server to complement Westwood's CAP client and support for other important emerging standards.

4. For the Pasadena and FutureReleases (Chandler 2.0+), we will focus on building Chandler to support third parties who will market the products for the mainstream, pragmatic audience. This will involve providing features that will help create and sustain a value chain for Chandler such as interoperability, migration, extensible framework, distribution support, technical support, and so on, as well as polishing and pruning the existing features of Chandler.

Table of Contents

I. Goals and Purposes of the Road Map

- Why a product road map?
- Chandler releases and naming conventions
- Open source and the road map

II. Product Vision

III. Theory of Adoption

- The Early Developer Releases
- Canoga End User Releases
- Westwood
- Pasadena

IV. Design Principles for Feature Prioritization in Canoga

V. 0.1 Release

- Goals and Purposes
- What's in it?
- What's not in it?

VI. 0.2 Release

- Goals and Purposes
-

I. Goals and Purposes of the Road Map

The Road Map is a strategy and planning document for Chandler. The ultimate goal is to provide a well-articulated step-by-step plan that if *well executed* will help fulfill OSAF's mission to gain wide adoption of Chandler.

This is a living document which will become more concrete and offer more details over time. Initially, we will focus on near-term releases (e.g. 0.2, 0.3 release) and major milestones (i.e. 1.0 release and Campus Westwood release). Also, to be truly useful, this plan will naturally be revised as we learn from our past actions. We need to know our execution strengths and limitations in order to derive the right plan. Thus, a future goal of this document is to remove as a caveat the italicized "if well executed" in the preceding paragraph.

Why a product road map?

- As a **communication vehicle**, it lets our internal and external stakeholders know what the product plan of record is, so that we are all aligned with the same goals and priorities and focused on the same set of tasks at every stage of the product.

- It provides decision-making criteria for **feature prioritization**.
- It **defines the goals for each release** and hence what success means for each release.
- It defines the main new features and components of each release. This will be the basis for functional specification of each release.
- It provides an **objective reference** as to how well we are performing to the goals of OSAF and acts as a tool for evaluating if our assumptions and strategies have panned out or if we should alter or switch strategies. The road map may also illuminate other options and alternative strategies we may decide to adopt.

Chandler Releases and naming conventions

- Release 1.0 = Canoga
- Campus Release = Westwood
- Release 2.0 = Pasadena
- Release 3.0 and beyond = Valley

Open Source Development and the Product Road Map

Open Source development offers the chance for serendipity -- the materialization of valuable innovations from unexpected directions. We can't plan for this, but we'll certainly be attuned to emerging innovations that expand our view of the project. Similarly, we cannot predict or control the aspects of the Chandler project that will generate the greatest contributions from the open source community. However we know that good leadership goes a long way. By carefully explaining the strategies behind this road map, continuously soliciting and incorporating feedback, and providing effective development leadership we hope to develop a cohesive community that is actively involved in fulfilling the goals of the roadmap as well as developing new innovations.

II. Chandler Product Vision

See [What's Compelling About Chandler: A Current Perspective](#) on our website.

III. Theory of Adoption

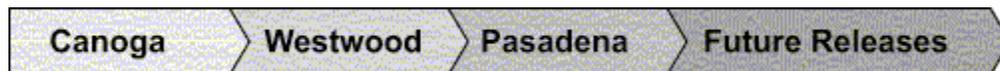
OSAF's mission is to create and gain wide adoption for software applications of uncompromising quality using open-source methods. Clearly, gaining wide adoption of Chandler is a key reason for Chandler's existence. Thus, we should develop a strategy for how Chandler will be widely adopted. This strategy will be a key driver to how we build Chandler and thus a key shaper of the product road map.

The market that Chandler serves, the (inter-)personal information management (PIM) market is extremely broad and appeals to large numbers and diverse types of people. In particular, almost all PC users have a need for email and, to a less extent, calendaring functionality.

However, given how broad Chandler's ambitions are and how wide the audience is, it is impossible to expect that we can build Chandler in a matter of months and expect the broad market to switch to Chandler, especially since there are several existing and well-entrenched solutions available today.

OSAF also believes that there are dramatically different and significantly better ways to serve the PIM market. In particular, there is an opportunity to offer discontinuous innovations in the way messages, events, tasks and other pieces of information can be organized and presented at the right context for people to efficiently get things done. We also believe that the sharing and collaboration of such information is an area that is ripe for innovation.

To introduce discontinuous innovations, we believe that there are several distinct stages with respect to product development.



In our cases, we call these stages: 1) Canoga Early Developer Releases (Chandler 0.2-0.5 release), 2) Canoga End User Releases (Chandler 0.5-1.0), 3) Westwood (Campus Release) 4) Pasadena (2.0). These stages correspond to the interesting parts of Technology Adoption Life Cycle (as developed by Harvard University and dramatically improved by Geoffrey Moore and the Chasm Group). At each stage of product development, we need to identify our target customer, their compelling reason to use Chandler and what the complete set of product and services are in order for the customer to fulfill their compelling reason to use Chandler. In later stages, we need to design Chandler to support partners and allies and other channels of distribution for Chandler.

Between Canoga and Westwood lies what is often called the Chasm. The former releases are intended for technology enthusiasts and early adopters. These are very different types of people (and look for very different things in Chandler) from the much larger set of people needed to be targeted next according to technology adoption life cycle - the pragmatists.

A way out of the Chasm is to target a series of very specific subsets of pragmatists in the next series of releases. This is known as the bowling alley strategy (where each specific segment of pragmatists is a bowling pin). Dominating the first few bowling pins help creates a virtuous cycle, allowing subsequent bowling pins to be 'toppled' with increasing ease. The Higher Education market can be likened to Chandler's bowling alley; specific bowling pins could be large research universities or smaller liberal arts colleges.

We describe each stage of the life cycle in more detail below:

The Chandler Early Developer Releases

Our initial 'dot releases' (0.1-0.5) will be targeted primarily to open source developers as there will be little to offer end users. In these releases, we will focus on designing and building:

1. **Platform and Infrastructure** (e.g. repository, storage, viewer parcel, application framework, etc.). Open source developers need a minimum set of

- supporting platform functionality in order to develop their own functionality. We also want to reduce reimplementations of major chunks of our product just because we did not cater for basic infrastructure e.g. Internationalization. Thus, we want to build our software with such infrastructure included as early as possible.
2. Our **landmark compelling "must-have" capabilities** will be discontinuous innovations of Chandler and will require multiple iterations of testing and usage to determine the right design (e.g. data model, Agenda-like Outline Table Widget, Sharing/Collaboration features). Because these features will take time to get right, it is better to get them right earlier in order that their changes will perturb the code less in the later releases. These are the features that will drive adoption of Chandler and we want to get feedback about these features as early as possible.
 3. A **modular or extensible architecture** to allow open source developers to contribute modules and code in a more loosely coupled fashion. This will offer some protection to third-party open source developers to reduce changes required in their code as we continue to make drastic changes to Chandler. For the APIs we unleash in 0.2 and 0.3 release timeframe, our goal is to have good stability for these APIs by the 0.5 release time frame. This means that after 0.5 release developers should be able to develop parcels with few changes needed for 1.0. Backward compatibility of parcel APIs is a goal we should strive for but is probably too early to decide when we can provide backward compatibility.
 4. Continue to provide basic elements of **end-user applications** (email, contacts and calendar) to drive and validate the design and testing of the above 1),2) and 3). In general, we believe that Chandler is both an application and a platform. We need to get the basics of the platform out first, but we need compelling applications to drive the definition of the platform. Thus, we cannot neglect one and focus exclusively on the other at any given release.

Canoga (End User Releases)

As we move towards the Canoga release (0.5 thru 1.0), our focus shifts significantly to our first target end user: the **info-centric early adopter**. Our organizational focus will be peer-to-peer (largely decentralized) groups of such info-centric users.

As the term 'info-centric' can be rather generic, we refer to an info-centric user as one with the following important characteristics:

1. Has a high need to create, process and organize information spanning multiple topics or domains and *inter-related* in many interesting ways.
2. Handles a *high volume* of information transactions on a daily basis (e.g. >25 emails/day and complex scheduling needs)
3. Is a self-declared *technology enthusiast*. This user is excited and technically able to try out new technologies. A simple statement of the benefits of Chandler is sufficient to motivate this user. It won't be necessary to rely on expensive 'push'-style marketing (such as brand, emotional or lifestyle marketing) to convince the user to try out Chandler.
4. Requires *little reliance on organizational infrastructure* to enjoy Chandler. For example, the user is either not tethered to a proprietary email or calendar

- system such as Exchange or Notes or is willing to use Chandler for tasks that are decoupled from such centralized infrastructure.
5. Has a need to *share and collaborate* with like-minded info-centric individuals in ad-hoc and informal groups.

The goal for Canoga then is to provide the minimum set of functionality that will convince and satisfy the info centric user to use Chandler. Thus, the main focus areas for Canoga are:

1. **Strong information management capabilities.** Internally, OSAF refers to this as the "soul of Lotus Agenda". We will provide an extensible framework for users to categorize, organize and retrieve all types of useful information such as URLs, attachments, notes, RSS feeds, etc. in addition to basic PIM data types such as messages, events, contacts and tasks. This generic information type (or Information Item as we call it) will have many built-in and extensible behaviors. For example, any Information Item can have arbitrary ad-hoc attributes added to it. Users can also relate information items to each other in user-defined schemas.
2. **Powerful outline view widget.** This is a UI widget that is capable of querying the appropriate set of information items and displaying the relevant attributes and relationships of this set of items so as to provide the right context for the current task at hand. For example, the user can view a project view that displays all the messages (threaded topically), events, tasks and documents related to a certain project.
3. **Power email features.** These are features to help the user process and organize messages faster while shielding the user from unwanted or irrelevant information (e.g. spam). For example, we will add capabilities to have one-click disposition of email messages under a variety of circumstances. This is not a kitchen-sink approach. We will not add all possible email features or even all features common to popular existing email clients (e.g. support for return receipts and possibly even signature templates for emails). We will focus on features dealing with high-volume email.
4. Continue refinement of **basic calendar and contacts functionality**, focusing on how calendar and contacts inter-relate with each other and with other information items. This will further drive and validate our information management features
5. Provide **sharing and collaboration** features around information items (this needs to be further designed and elaborated)
6. **Robustness and reasonable performance.** Canoga will be our first release that should be safe for real data. We will have assurances about data integrity and recoverability. We will also ensure that future versions of Chandler will seamlessly accept Canoga repository data. We also need to ensure reasonable performance since we are targeting info-centric users who will process and store much more information than the regular user. Our goal is allow users to store and interact with a few million Information Items.

Westwood (Campus Release)

As we move beyond Canoga, our focus shifts to ensuring institutional or organizational adoption of Chandler. We have identified the broad Higher Education segment as our likely first institutional adoption target. This is the target market for the Westwood releases.

A principal segment (or bowling pin) within Higher Education is the large research universities. OSAF is currently evaluating the requirements of this segment as part of a grant provided by the Mellon Foundation.

The initial Westwood release will be focused on providing the minimum set of functionality that will enable widespread deployment of Chandler in the Higher Education landscape. Such functionality will likely include:

- Central servers to ease central administration and to support nomadic access requirements where students may require access to information from multiple computers
- Richer and standards-based calendar functionality including CAP client compliance
- Interoperability with existing university infrastructure such as LDAP servers, authentication and calendaring systems, etc.
- Higher level and university-specific security and privacy issues

Another likely segment within Higher Education is the far smaller liberal arts colleges. We are also evaluating their potential requirements which will likewise require interoperating with their existing infrastructure. Westwood requirements will be developed and described in more detail as part of our deliverables to the Mellon Foundation (in a separate document).

Pasadena (2.0)

In Pasadena, we will focus on providing support for third parties to customize Chandler for Small and Medium Businesses and other broad targets. OSAF does not intend to provide direct support for any of these large constituents. Instead, we are hoping that third parties will arise to take on these challenges. Thus, OSAF does not intend to take on a for-profit model like other open source organization such as MySQL or JBoss. Instead, we hope that several 'Red Hat'-equivalents will emerge to take on the challenge of marketing Chandler to the mainstream audience.

As we move towards attracting the mainstream, pragmatist audience for Chandler, we face new challenges:

1. Pragmatists want proven solutions with *happy, reference-able customers* that are similar to themselves. Hopefully, our success in Westwood will be sufficient to overcome such objections
2. Pragmatists require simple deployments requiring little customization to *integrate with their existing infrastructure*.
3. Pragmatists require *seamless support and hand-holding* before they can become comfortable with the new technology
4. Pragmatists require more 'push' marketing to be convinced about Chandler's benefits. This implies the emergence of *third-party (i.e. non-OSAF) sales, distribution and marketing channels*.

From the product perspective, to meet these new challenges we should focus on:

1. **Interoperability**. We should provide a framework and actively encourage the open source community and other third-parties to ensure that Chandler

- works and play well in as many environments as possible. This may also include support web and PDA clients
2. **Migration and the Initial Experience.** We will make transitioning from an existing PIM or email client seamless. This will require painless migration of a user's existing data over to Chandler. It also means supporting the more peripheral features that users of other clients have come to expect (e.g. return receipts)
 3. **Extensible framework.** We should provide functionality in Chandler to nurture a value chain so that it can be easily customized and extended by third-parties for a diverse set of market segments. This calls for an extremely easy but powerful extensible framework
 4. **Distribution.** We should remove major encumbrances to distributing Chandler. This requires understanding the requirements of distribution channels such as RedHat, PC manufacturers, VARs, etc.
 5. **Support.** While Chandler should be extremely easy to use, where necessary we will provide a sophisticated, customizable help infrastructure to allow Chandler users to help themselves. We will provide tools for our third-party value chain to make support of Chandler for their customized purposes even simpler.
 6. **Polishing and Pruning.** Finally, there will be many lessons learned from the Canoga and what we learn.
-

IV. Design Principles for Feature Prioritization in Canoga

The rest of the document will focus on releases towards Canoga, as these are the releases of immediate importance. The following are design principles we will use to prioritize tasks as we progress towards Canoga:

We are **innovative-landmark-capability-centric**. Innovative landmark capabilities are the pivotal features that will compel a user to switch to using Chandler. There are no identical features available to emulate on other clients and hence require many iterations to design, refine and get right. The earlier we seed these features, the more early adopters we attract, the more feedback we receive and the faster we make progress on the design. Thus, we need to start developing these features first.

We have identified four areas of innovative landmark capabilities to focus on:

- **"Soul of Agenda".** We want to allow users to organize information the way regular people think about and act on such information, not the way applications are structured today. Thus, we want to allow users to create attributes for information items in an ad-hoc fashion, allowing information to be structured and unstructured at the same time. Users require different items and different attributes of items to perform different tasks. The goal of this feature is to provide just the relevant information for the user to perform the current task structured (displayed) in the appropriate manner.
- **Sharing and Collaboration.** Many tasks today involve cooperation with a whole group of people. What information should you share with your collaborators to get the job done? Chandler endeavors to provide the framework for such information sharing and collaboration. More concrete specifications will be described in a separate document on sharing.

- **High-Volume Power User features.** Our initial target user will have a high volume of information transactions. We want to provide the tools to process such transactions as efficiently and effectively as possible. This includes dramatically faster and more flexible searches to refinements and polishing of current UI such as a one-click disposition of email messages for a variety of common reasons.
- **Extensibility.** We want developers and end-users to be able to customize Chandler to their utmost satisfaction. At the minimum, this entails allowing sophisticated filters on views, and allowing queries on views to be persistent. We want to provide for end-user scripting to redesign Chandler to one's liking and to allow for hooks into the Chandler event model. Ultimately, we want to create a agent-framework that handles routine multi-step operations seamlessly on behalf of the user. We believe that many of these extensions can be shared for widespread usage.

We will provide a **strong foundation platform and infrastructure.** As much as possible and relying on our past software development experience, we will try to build the foundation for Chandler such that future functionality will not call for reimplementing major chunks of the product. e.g. we will think through Internationalization issues now rather than build 1.0 and only then start to attempt a localization of Chandler. Providing a strong platform first will, hopefully, also attract open source developers earlier in helping to build Chandler.

We like **Modular frameworks.** Such frameworks should offer rich functionality and be loosely coupled with the rest of Chandler such that developers can be shielded from changes in other parts of Chandler. As much as possible, we want to develop well-thought out frameworks early so that developers can start contributing to Chandler in parallel and yet not have to rewrite major sections of their code as Chandler matures.

Chandler offers **the duality of being both an application and a platform.** We believe we cannot create a viable platform without testing it out on a series of non-trivial applications. Thus, applications drive the development of platforms. We need to build both applications and platforms more or less in parallel.

Breadth-first. The above principles imply that we will implement features in a breadth-first fashion. e.g. We won't implement only calendar features for Canoga. Instead, we will implement a broad foundation and platform for information management and focus on email, calendar, tasks, and notes as specific applications. We will lead by demonstrating the innovative landmark capabilities in each application area and hope that the community will build on top of our frameworks to make the application a usable whole product for different communities.

Bottoms-up Scheduling. This road map is not yet ready to provide specific dates for the releases we are describing. We believe that we need to do bottoms-up scheduling and resulting trade-off decisions before we can arrive at a realistic schedule.

V. The 0.1 Release

The [0.1 release is described](#) in detail on our website.

VI. 0.2 Release

We will be publishing more specific information about our forthcoming 0.2 release in the next few weeks.