In A Nut Shell

Duration: 90 Min.

Session High lights:

An experience report on building a large scale virtual university platform iteratively using agile discipline and techniques

Presents architecture and design techniques to use open source platforms in the large scale

Exercises and insights on using the principles of collaborative computing and Schema–less databases for modeling flexible workflows and rapid integration

Whirligigs and Mountains

Agile Product Development in a Large Enterprise

Summary

We present our experiences of using agile discipline that enabled the success of a technology enabled learning product within India's largest private education enterprise.

We present lessons learnt in overcoming challenges of iterative product visioning, in transforming technology unaware stakeholders into product champions and in developing multiple iterations of several variant products on a common code base while working with a small development team. We present architecture and design principles that enables us to transform from programming in the lab to the programming in the world.

Background

Manipal Education, where this story is situated, is India's largest provider of education in the private sector with over 220,000 students, three universities, one very large distance learning department, more than 30 institutions and over 700 subjects. Four years ago, we signed on at Manipal Education to - as defined by CEO Anand Sudarshan - "figure out how technology can enable learning and help us scale lock in step with quality". However, beyond this it was greenfield territory - the

"EduNxt today is used by over 220,000 students and operates at one tenth of the cost of ownership of comparable commercial offerings"

domain of large scale higher education, especially distributed learning, was (and still is) emerging.

It was in this context that we set out to develop EduNxt that today is used by over 220,000 students and operates at a one tenth of the cost of ownership of comparable commercial offerings from world leaders in the learning technologies space. We had no choice but to adopt an iterative development model and this has led to significant successes over the last few years. The infographic below summarizes the key stages of this journey.

	2008 when we started	April 2009 - the inflection point	October 2011 - today
The vision of the product	As an optional add-on to support existing distance education. Provide course material and online quizzes. Enable Course Management	Key component to be used in all distance learning programs and also to be introduced into campuses. Enable a Virtual University	Key component of all degree based and vocational learning. Offer as a service to other institutions. Support Multiple Universities in a SAAS mode.
Stakeholder involvement	Minimal, uncertain about roles and how to use the technology, localized support.	Increased enthusiasm, emergence of product champions amongst stakeholders	Broad based support, emergence of product development boards
Approach to funding and organization commitment	Small budgets, a wait and watch approach	Guaranteed 3 year budget, strategic project across the organization	Long term funding, organization sees it as monetizable intellectual property

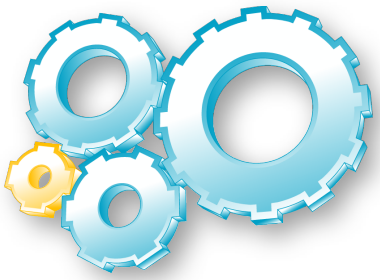
	2008 when we started	April 2009 - the inflection point	October 2011 - today
Approach to technology and development	Existing open source LMS tools like Moodle. Minimal customization, rapid releases of prototypical features almost on a daily basis.	Combination of open source tools and custom developed software. 3-month product release calendar with multiple smaller interim releases.	Combination of open source tools and custom developed software. Separate core and product specific development. SCRUM based approach extending to both core and customer specific development
Team sizes and organization	4 PHP developers, 1 lead and 1 architect	8 PHP developers, 3 leads, 1 architect	8 PHP developers, 3 leads, 1 architect, 4 person testing team, 3 customer representatives and 1 project manager

Key Learnings

Leverage open sources technologies: - while the evident benefits was the reduction of licensing costs, the greater benefit of open sources technologies comes from the way it improved our development productivity. Perhaps a decade and a half ago, to use open sources one needed to have what we would call "super-cat" programmers, who could use undocumented features, figure out obtuse code and make things work. Open sources technology today has come a long way and has enabled "programming in the world" - almost anyone can work with them, several tools do not even need anything more than minimal customization to use. Using open sources has enabled us to get around the challenge of hiring top-quality development talent into a non-IT industry organization. Another way of looking at it is that in addition to our 8 person development team, we have a very large community with several expert programmers building features that are useful to us.

Prototype often, but Production Quality Always: There are no well defined business/domain specification for the education and learning

management systems. And, there are no technology stacks and platforms that can scale to meet any learning management system's demand. Therefore, the only way for all stakeholders to create a shared understanding of what should be and what could be built is to quickly ideate on a set of functionality, and then rapidly build it, release it and use the feedback to derive the next set of features. However, the rapidity of the feature building cannot be done at the cost of the design and architecture efficiency. We made some key architecture and design drivers that enabled us to reuse everything we built.



Open sources technology today has come a long way and has enabled "programming in the world"

Prototype often, but Production Quality Always

Anchor the design on variables that change less often

Iteratively increase the complexity of the architecture

Anchor the design on variables that change less often - one of the downsides of a purely iterative and prototype led approach was the absence of consistency in the design and implementation, redundant code and a lot of confusion at the developer level on what needed to be done. The net result was purely functional implementations with functionality being translated straight through into code and precious little concern for quality attributes (other than performance). Once we learn this lesson, we started identifying those key domain objects that formed the base of our technology and could be combined and configured to product individual features. Over a period of time we have got better at this and today have a common code-base shared across products with fairly well-defined variation points.

Iteratively increase the complexity of the architecture - once in a while a new idea comes along that refines our understanding of the domain itself and questions the fundamental building blocks we have developed out system on. We have refactored our architecture and design twice now in journeying from a purely course centric view to a virtual university view. In this process we realized that architectures can also be designed iteratively, and each version of the architecture has to be designed on top of the previous version - in this fashion, we were able to reuse the earlier versions of the product as building blocks for the next version.

Session Takeouts

- How does one develop a technology roadmap, especially in an area that is in itself evolving or if one is in a startup product company developing a emerging product. In a way we are trying to bring in a

degree of predictability into something that looks potentially unknown. We explain how to dimension the problem into manageable chunks, how to look for and use standards or reference implementations.

- What are the techniques of architecture and design, how do we factor for quality attributes such as extensibility, modifiability and so on, where do these come in in the development lifecycle. The pitfalls of a purely functional approach and why we emphasize that an agile team must always build features for itself before it can build features for its customers.

- Agile project management methods that worked and those that did not work for us, how to bring clarity to the team while managing evolving functional requirements, how do we keep the team informed on status, what do we do in the scrum meetings, suggestions from the team that helped us improve project management such as tracking technical debt.

- How did we improve the quality of our code through automation of engineering processes, how did we conduct the various refactoring exercises, when did the clash of promised functionality versus need to refactor emerge and how was it addressed.

Exercises and Participant Interactions

- An application of the Supportability Framework presented by us at the WICSA conference at Boulder, Colorado in June 11. We illustrate how to use the supportability framework to identify concerns of stakeholders such as development and operational teams relating to quality attributes such as extensibility, modifiability and so on and then how to identify features to be added to the system that address these concerns, thereby supporting the stakeholders.

- Demonstrate how collaborative computing techniques are being increasingly used to overturn our traditional understanding of workflow systems in which process subsumes content to create agile workflow environments where the workflow is embedded in the content thereby enabling systems to scale and be modified without re-development.

- Demonstrate how we are using schema-less databases and RDF based integration of open sources technologies to reduce integration effort and enable scale.

Speaker Profiles

Satish Sukumar

Satish has more than 17 years of experience in the IT Industry. He held various technology, development and support positions. He started his career with Microland and later he was the Head of Technology for Planetasia, Vice President of Engineering for Veloz Global Solutions, India.

He consults as CTO for the Manipal Education. He is a well sought after speaker at CSI, EDGE and other industry forums. He was featured on the cover page of CIO magazine. He has several patents and publications to his credit.

He is the co-founder and chief technology consultant at Canopus Consulting, Bangalore.

He can be reached at satish@canopusconsulting.com

Nagaraju Pappu

Nagaraju has about 20 years of experience in both academics and industry. He held various research and engineering positions both in India and in USA. He started his career as a research engineer at IIT-Kanpur and he was later with Oracle India, Fujitsu Software Corporation, InfoDream as senior engineer and architect. He specializes in large scale transaction processing, semantic computing and software architecture. He is a pioneer in cultural informatics in India.

Nagaraju has several patents and publications to his credit. His papers are published in top-tier international journals and conferences. He has conducted several public and academic courses and is a highly sought after speaker at CSI events, CII and other industry forums.

He is a visiting faculty to IIIT-Hyderabad and IIT-Kanpur. He is the co-founder and chief solutions architect at Canopus Consulting.

He can be reached at pnr@canopusconsulting.com