

On the Radar: BigLever Software Inc.

Vendor briefing note

Bola Rotibi

April 2009

This *On the Radar* vendor brief follows a simple “ten questions” format, which we designed to provide a concise but thorough overview of a company and its products and services. We use this format to focus on the capability and suitability of small, specialist vendors – to help you build the best possible vendor shortlists when looking to make new technology investments.

Macehiter Ward-Dutton is a specialist IT advisory firm which focuses exclusively on issues concerning **IT-business alignment**. We use our significant industry experience, acknowledged expertise, and a flexible approach to advise businesses on IT architecture, integration, management, organisation and culture.

www.mwdadvisors.com



Licensed under a Creative Commons License: www.mwdadvisors.com/about/open.php

1. Who?

BigLever Software Inc. (www.biglever.com) is a privately-held company that provides the engineering tools, framework and services for establishing “software product lines”. The company is headquartered in Austin, Texas in the USA.

2. What does it do?

Software Product Lines (SPL) is a mechanism and a methodology approach for delivering feature variations of software-based products and applications and managing the evolution process productively and repeatedly. It emphasises a way to construct and manage software assets – such as source code, requirements, architecture and design models, and test cases – for intentional reuse from the very start. It employs the proven manufacturing processes and best practices of automation and production line delivery to assemble software assets into different feature variants of a software-based product or application. Moreover, SPL supports the common themes impacting the software delivery process (e.g. MDD, Agility, Test and Requirements-driven development etc.) and consolidates them into a pragmatic, workable, user-specific methodology and process for achieving greater reuse across a product line and better software delivery success.

Ultimately BigLever’s tooling portfolio is designed to implement and support Software Product Lines.

In BigLever’s view, taking a product-centric focus offers little or no consideration for easily supporting variations of the product. It forces one to manage and coordinate the often complex inter-dependencies and labour-intensive interactions required to implement or to take advantage of the commonalities found in the software assets used to deliver a particular software product so that they can be configured and reused to create and maintain variants of the product. The result of taking such a product-centric approach raises both the development cost and complexity stakes.

The direction and strategy of BigLever’s tooling and framework is a move away from a product-centric focus – where the concentration is on effectively managing the lifecycle of a single software-based product or application – to one that focuses on a “single-system” perspective so that it can be used to replicate variations of the product according to a specific feature profile. The single system in this case is the production process required to deliver a software product.

This focus on the production process is the central tenet to recreating multiple variations of a specific software product or application, since the production process remains the same in all the variations whilst the feature profiles differ. The software production process is an abstraction of the software delivery lifecycle complete with the policies and rules for governance to a desired outcome and goal.

At the centre of BigLever’s portfolio is the Gears Software Product Line (SPL) Lifecycle Framework which provides the single-system perspective of the software production process for the common software constituents and assets of a software product or application family. The framework provides a common set of constructs that support the SPL process for all tools and assets and enables an orderly transition through each phase of the software development and delivery lifecycle. The individual products in the family group, defined by their feature variations, are then produced by automating the assembly and configuration of the common software assets according to specific feature variation profiles.

The Gears SPL Lifecycle Framework consists of three core components for establishing and maintaining an effective and efficient software production line platform and process:

- **Software Assets** – These include requirement specifications, architectures, models, designs, source code components, test cases and documentation. All are considered reusable as they can be configured and composed in different ways to create all the variation instances of software assets in a particular product family line. The variation points are implementation-specific differences among the products in a portfolio, e.g. background colour, memory size, internationalisation and localisation features or the positioning and structure of a logo. The Gears framework uses a proprietary language to code each variation point with the intelligence to configure itself based on a product feature profile.



- **Product Feature Profiles and Models** – These describe the optional and variable features for the software products in the product family line. Each product within the product line is uniquely defined by its own feature profile. Through the Gears Feature Model language, software architects can create models that encapsulate at the domain level all the feature variations that make up the different software products within a family product portfolio (or product line).
- **The Gears SPL Product Configurator** – This is the engineering tool that automatically assembles and configures the necessary software assets (source code, test cases, build descriptors etc.), guided by the product feature profiles, to produce the individual products in the product family portfolio. The configurator configures the variation points to match the feature profile of the individual product.
- **Gears Development Environment** – This provides the workbench/console for developing and delivering the different portfolio of products. Like many other integrated development environments, it provides a collection of browser interfaces, editors, dashboard views, wizards and analytical tools to simplify the delivery lifecycle for product portfolios. Typical of today's integration requirements, the Gears IDE supports integration and interoperability with existing development tools investments as well software already deployed within the organisation.

In addition to the Gears SPL framework, BigLever also provides a three-tiered SPL methodology built on the back of simplified patterns of successful customer deployments. The three-tiered model is aimed at helping users to understand the value proposition of a Software Product Line process and recognise the benefits from implementing a SPL framework and approach to delivering and maintaining family product portfolios. The tiers are structured as follows:

- **Base tier: Variation management and automated production** – Here the focus is firstly on the creation and management of variation points within software assets, and then secondly on delivering the automated processes of the Software Product Line using existing tools and assets.
- **Middle tier: Core asset-focused development** – The focus here is twofold: the first is on developing the core assets that make up the “single-system” perspective of the product line production process; the second is organising the team for SPL asset-focused development rather than product-centric development.
- **Top tier: Feature-based portfolio evolution** – The focus here is on transitioning the “business to feature”-based portfolio evolution where simply modifying the feature requirements and variation points leads to the delivery of a new product in the family portfolio. This high level tier looks to streamline the communication and collaboration between the various engineering and delivery teams in order to reduce the time to market.

3. Who is it for?

The fundamental approach to managing feature variations and in automating the delivery of individual products within a product family group means that BigLever's Gears tooling and services portfolio is suitable for those organisations required to deliver closely-related products. These characteristics cover a broad set of organisations spanning multiple industry sectors.

More specifically, organisations and software delivery teams who will find BigLever's Gears portfolio and the Software Product Line approach to developing and managing product family portfolios suitable will be those looking to:

- build new software products that have volatile product concepts and potentially unstable domains as teams look to venture into new areas with prototyped solutions;
- provide mass customisation of products and services;
- offer feature variations in their product sets like internationalisation and localisation (I18N), physical attributes, functional capabilities, porting and deployment differentiations;
- port applications, especially where the porting process includes the addition of new features.

BigLever's own client portfolio spans aerospace and defence, automotive, computer and data storage, consumer electronics, medical, online retail, telecommunications and R&D. Usage scenarios include those developing software that is embedded, enterprise, stand-alone, web-based or service-oriented.

Ultimately, the Gears platform is designed to support the individual roles responsible for the key phases of the software production process in identifying and configuring the variation points within their respective functions that will determine individual products within a product family portfolio. This makes Gears a target for roles such as requirement engineers, architects, modellers, developers, build engineers, document writers, configuration managers, test engineers and project managers.

4. Why is it interesting?

The value proposition of BigLever's Gears tooling portfolio comes at a time when there is increasing reliance on software components within manufactured products coupled with a sizeable and growing market for commercial software applications.

Few organisations creating software-based products and applications – either for commercial purposes or for internal customers – do so as a single one-off, stand-alone product. More often than not there will be multiple variations of the product, where the variations will be determined by features such as internationalisation and localisation requirements, multiple power and performance requirements, function capabilities or physical attributes etc.

The challenges facing organisations charged with delivering variations of software-based products and applications are ones of consistency, repeatability and confident and productive reuse. How you bring together reuse across multiple different instantiations of a software application or component within a product is difficult and complex. Productive software reuse that doesn't put a strain on the software delivery team's ability to deliver quickly or drain vital resources has been a challenge for many years. For many organisations it has been an unattainable goal without incurring costly side effects.

Moreover, the production-like requirements of delivering software-based products and commercial applications, sees a convergence of the concerns of the software delivery and management lifecycle process with those of the product lifecycle management process typical of manufactured goods. This push towards industrialised and production-like software delivery is one of the driving forces behind the SPL process, which focuses on automation, the pragmatic and intentional reuse of software components and managed repeatability. SPL is fast becoming recognised by many of those responsible for delivering software-based products (along with key institutes like the Software Engineering Institute - SEI: Carnegie Mellon) as the most appropriate framework for bringing together and managing the convergence between the software delivery and product lifecycle process productively and repeatedly.

However, there are few tools that directly support Software Product Line delivery processes, although many of the tools and platforms used to support the software delivery process (e.g. requirements capture and management, modelling and source code editors, change control and source code configuration management, build and release management) can be configured to support SPL by those experienced and skilled in the SPL mechanism and processes.



Therefore BigLever's Gears tooling platform is particularly interesting for the following reasons:

- It has been architected to directly support the Software Product Line mechanism and methodology approach, thereby considerably simplifying its implementation and making intentional software reuse a pragmatic and realistic goal;
- It has an integration and interoperability framework designed to support many of the existing tools used within the software delivery process particularly those typified by the Application Lifecycle Management (ALM) framework, which not only simplifies the implementation process, but preserves existing investments and supports the delivery team without detracting them from their normal workflow. That said, what is still required for success, is a level of commitment to shift existing processes and practices to become more aligned to those of SPL;
- The ROI figures and operational metrics produced by early adopters of SPL and BigLever's own client base are impressive and are of an order of magnitude that give rise to the kind of productivity and business value returns that organisations would appreciate and greatly desire;
- It intrinsically leverages common practices such as Agile development processes which are being eagerly adopted throughout industry;
- It reinforces the discipline and governance processes of ALM and PLM without impinging on either's dynamics;
- It is a tool that embodies the principles of software and systems engineering, designed to support the skills and working practices of such roles without overburdening them with excessive processes.

5. How established is it?

BigLever Software was founded in 1999, based on 12 years of leadership in Software Product Line R&D. Headquartered in Austin, Texas with a staff of over 15 employees, contractors and consultants, it has been revenue-funded and profitable since inception. Revenue growth has been consistent and the booked revenue in Q1 and Q2 of 2009 exceeds total annual revenue in 2008.

BigLever has been selling Gears commercially for the last eight years. Its customers come from a range of industry verticals such as aerospace and defence, automotive, computer and data storage, consumer electronics, medical, online retail, telecommunications and R&D. They include Lockheed Martin, LSI, HomeAway, Honeywell, FKI Logistex, Roche Diagnostics, Caterpillar and FujiXerox.

6. How open is it?

BigLever's Gears tool and framework supports a comprehensive integration interfaces the framework API and various Gears bridge connectors for existing tools and assets, such as:

- Programming languages and compilers, e.g. Java, C, C++, C-Sharp, Ada, Perl, XML, HTML etc.
- Integrated Development environments, e.g. Eclipse, Visual Studio, SlickEdit, GNAT Pro.
- Requirements management systems, e.g. DOORS, Requisite Pro, Calibre.
- Configuration management systems, e.g. Synergy, ClearCase, Subversion & Perforce.
- Build tools, e.g. ANT, Maven, gmake, nmake.
- Model-Driven Development (MDD) tools, e.g. Rhapsody, Artisan.
- Documentation tools, e.g. Word, FrameMaker, Pages.



7. Who does it partner with?

BigLever has a broad set of partnership relationships with ISVs, VARs and distributors, including IBM Rational, Noblestar, EVOCEAN and CTC, who provide tooling services addressing parts of the software delivery process. It also partners with members of its client base who have contributed plug-ins and add-ons that have extended its Gears platform.

BigLever partnered tightly with Telelogic. Since the latter's acquisition by IBM and incorporation into the IBM Rational group, BigLever, has strengthened its partnership with IBM Rational, joining forces to deliver a joint marketing and sales campaign to raise the profile of SPL and the combined capability of their tools and services experience in this field.

As mentioned above, the company provides a number of Gears Bridge connectors that integrate tightly with the IBM Rational Products enabling it to support the "Ready for IBM Rational software" validation.

8. Are there areas for improvement?

Although BigLever's CEO and founder Charles Kruger is behind much of the content on the official Software Product Lines website, BigLever has struggled to get the SPL message and the success that both the process and its Gears tools platform has achieved to a wider audience. Despite being able to impact positively on valuable business metrics (such as reduced costs, improved quality of products, discontinuous jump in productivity and innovation capability, and faster time to market etc.), much of the problem has been down to simplifying the messaging for a tooling product that targets the heavily technical and product-oriented world of software and systems engineers. The technical nature of SPL makes it a challenge to hold the interest of those less inclined.

Sales cycles are typical for missionary sales of a paradigm shift by a small company – i.e. very long. The partnership with IBM should improve this and the wider marketing challenge.

Another issue has been in providing easily accessible and simple demos that can show the speed at which Gears can be implemented and made to start managing feature variations and automating the creation of different products within a product family portfolio. Many of BigLever's customers believe Gears to be an easy tool to implement but it could benefit from being made available to a wider audience in much the same vein as an ANT build tool is easily accessible. A cheaper version that limits the number of variation points would also lower the barrier to entry and potentially broaden its appeal to a wider audience.

One possible area for further focus would be in promoting the software reuse capabilities in a more consumable manner to enterprise IT delivery teams. Providing better alignment with this group offers good potential since many of the reuse challenges and requirements apply. The difficulty for the company will be in catering for the needs of a group that is not always driven or governed by the disciplined processes and environments found with those actively engaged in Product Lifecycle Management (PLM) processes. Demos showing how SPL can be used to support service-oriented applications or applications that might implement Web 2.0 content, for example, could help to raise the relevancy of the framework in this market.

Whilst BigLever provides a methodology support program, more could be done in engaging with professional services partners who can help to broaden best practices. The company could also do more to link the metrics and success stories found on the official SPL website with those of its own customers to offer broader insight into what can be achieved for the business and the kind of business metrics and benchmarks that make sense.



9. What's next?

Part of BigLever's challenge is in getting buy-in from those with the power and commitment to champion the value proposition of a Software Product Lines approach. This requires a person with senior authority who is technically adept and also has the commercial acumen to fight the technical, cultural and financial arguments. Once this person is in place, the process of adoption will then need to be supported with "point and click" tools that maximise developer productivity with little effort on their part.

However, to combat this challenge and raise the profile of its portfolio, the company's roadmap plans include the following:

- Based on growing experience, continue to expand and standardise the Gears SPL framework API to amplify tool and asset integration, and increase the number of off-the-shelf integration bridges for commercial tools.
- Continue to extend the Gears Development Environment with advanced capabilities for comprehending, analysing, optimising and reporting on the global "health" of a product line.
- Establish baseline for rapid growth with enablement resources and training program for scalable direct and channel sales programs.
- Utilise thought leadership and informational marketing to expand mainstream industry awareness and standardisation. Establish SPL as a well-known and widespread industry practice.

This is a comprehensive list of next steps that will undoubtedly present a challenge for BigLever to achieve in a short time frame. However, the selling and marketing partnership with IBM will go some way to raise the profile of the technology and the company. Given the size of the organisation, the company would be well advised to seek the support of its community to help tackle some of the development challenges. Once again we see the benefits in this regard as a by-product of the IBM partnership.

10. Should I consider it?

As discussed in 'What does it do?', 'Who is it for?' and 'Why is it interesting?' BigLever's Gears tooling and services should be of interest to you if you have to develop and maintain portfolios of family-grouped software-based products and applications. Through its support for SPL, the Gears tooling allows you to plan for intentional reuse. However, even though proactive reuse is supported within the Gears framework, reactive influences such as those required in prototyping are also catered for. Gears should also be of interest if you utilise agile software development methodologies and processes and if you wish to manage feature variability over the entire software product or application lifecycle. The tight partnership with IBM Rational and the joint marketing and sales relationship will provide a level of service promise that should reassure those concerned with BigLever's size.