

Non Recurring Engineering

Daniel Shefer

Director, Business Development

SanDisk

DS_PM/delete-this/@spamex.com

What is Non-Recurring Engineering?

Non-Recurring Engineering (NRE) is a one time engineering effort by a vendor that is paid for by a customer. NRE is driven by a feature or capability that a product lacks and a customer wants enough to pay for.

The fundamental issue that NRE highlights is that the demand for it is triggered because the product doesn't provide the customer with enough value out of the box.

NRE is generally requested by customers under the following circumstances:

- A customer doesn't like the way the system behaves. This could be a missing feature, a platform that is not supported etc. The customer may look for a way to convince the vendor of the importance of their request. Offering to pay for NRE is one way to do this. A customer's willingness to pay for NRE is a good indication of how much they want this done.
- A customer wishes to move up the priority of an already planned feature into an earlier release. From the vendor's perspective this is basically a resource scheduling issue. The vendor needs to carefully examine the implications of changing their schedule.
- The vendor stopped supporting a previous version and the customer is willing to pay for making changes to this version.

"The good news about NREs is that when customers spend money on NRE they are planning on being a customer for a long time."

Luke Hohmann, Principal of Enthiosys and author of [Beyond Software Architecture](#)

What NRE Isn't

NRE is not a good way to finance core functionality. It should also not be used as a device to stuff things into a product that are not in the vendor's best interests in the first place.

NRE isn't a replacement for professional services either. Customization and integration, especially "one offs" are the domain of professional services, not development. The product architecture needs to support this and there need to be the resources and skill set to pull this off.

NRE is not a way to craft a product strategy. It should not be allowed to drive decisions that are not in the ISVs long-term best interests. If something doesn't make sense to be a part of the product, NRE shouldn't get it there. If the customer simply demands that it be done, hand them over to professional services or a custom development shop.

The bottom line is that NRE work should not be done for the core product or, on the other hand for something that is not in the path of the company's strategic direction.

Things to Consider Before Embarking on an NRE Project

- A major issue that is highlighted by requests to do NRE is not knowing when to say "no" to a customer. This has to do with many factors but at the end of the day there has to be someone in the company who has the willingness and authority to refuse to embark on such a project.
- Beware of an NRE project that is presented by Sales as a "deal closer." It takes a lot of discipline and skill to vet these "closers" properly. Discipline that Sales may lack. Not to mention that they are not compensated or promoted for holding back.
- NRE projects have a lot more impact on the product and the development process than meets the eye. Areas that can be impacted by an NRE project include:
 - Revenue. Work that is done on NRE is not being done for the core product. This represents lost revenue. Also, changing the timeline of planned features can affect the willingness of customers to buy or upgrade.
 - Customers' expectations. Customers can become very upset if features they were promised are postponed.
 - Integration with other modules and products. NRE projects increase the complexity of the product and the resources needed to support and upgrade it in the future.
- If the ISV's competitors have a feature that is desired by customers, they will be less inclined to pay for NRE as it is perceived in the vendor's interests to play catch up. The looming competitive gap is also used by customers as leverage for their demands.

Pricing NRE Projects

Now that you've decided to embark an NRE project, how do you calculate how much to charge the customer?

In theory, costing an NRE project is easy. Estimate the engineering hours and overhead costs, add a "fudge" factor for uncertainties and risks and multiply it by your grossed cost rate. The result should then be multiplied by a factor of 5 or so. This is assuming that your company spends 20% of its revenue on R&D¹.

It is critical to understand the reason for this "5" factor, especially when the NRE takes up a significant portion of engineering's resources. If engineering costs are around 20% of revenue, an NRE project costing X would have a 5x negative impact on sales. This is because the resources used for the NRE were diverted from producing sale-generating

¹ The [SoftLetter](#) Benchmark 50: Research and Development. These figures span 2001 through 2003. 20% is the software industry median. Desktop, Internet applications, and education companies spend the most, 24%, 27%, and 24.5%, respectively.

products. This indirect cost must be taken into consideration and not doing so runs the risk of embarking on revenue generating but money losing projects.

A company that is in good financial condition, i.e. not under financial duress, can follow this procedure for costing an NRE projects but a company is on the brink will probably not be able to achieve these premiums.

Support and maintenance costs needs to be priced as well. If the NRE is not incorporated into the product and is available only to the customer who paid for it, support is extra. These costs should be more than the going rate to compensate for the added difficulty of supporting code which is basically a one off. This is because each maintenance release becomes an NRE project in itself.

Who Owns the End Results of NRE?

When a customer pays for NRE, they expect to have rights to the functionality they paid for. There are several approaches to this:

- The NRE work becomes an integral part of the product. This is the easiest solution for the vendor. For traditional commercial software, this is the model that makes the most sense.
- Only the customer that paid for the NRE gets to enjoy the new capabilities. This is a costly prospect for the ISV.
- The ISV offers the customer time- limited exclusivity to the added functionality. This way, the vendor can leverage it for other customers. (This is basically a professional services model. The ISV's goal here is to "productize" the NRE and add it to the price list.
- Both sides agree on a revenue sharing scheme for the revenue generated from the new functionality. In this case, the NRE work must be packaged in a way that it can be sold separately. A downside for this approach is that it is hard to set up and maintain over time. Therefore, this agreement should be time-limited. Few companies and few customers are positioned to make #this work although chances are better if the NRE is done as part of an OEM agreement.

Additional Pointers on NRE Projects

- Another way to motivate customers to pay for the NRE is to bundle it with a discount on future purchases.
- One way to reduce the cost per customer is to get a consortium of customers to pay for the work. This is a difficult thing to pull off and customers might ask why they should pay for development if others need it as well. In their mind, it should come from the vendor's resources. And they would be right!
- Offer a limited warranty to the NRE work. Also, decide whether to fix bugs for free or to charge for that. The issue is that you just developed something that is not necessarily in your main line of business. In any case, enhancements cost.
- If the NRE work is something you wanted to do anyway, it's a resource scheduling problem not an NRE project.
- NRE almost always delays scheduled releases.
- The thinking "we will hire consultants" to do NRE work is mostly delusional. Even if you do hire outside hands to do the work, you have to be aware that every problem they have, every thing they are not 100% sure about, they will have to

stop someone's work to get an answer for. If the product's interface and API are not very well defined, hiring a consultant to do NRE work is basically like having to train a new employee.

- A patch that is made to an old version does not necessarily mean that the patch is applied to the new version.

I hope you found this article informative. I always enjoy hearing from my readers so your comments are appreciated!

Daniel Shefer

About Daniel Shefer

Daniel Shefer is Director of Product Architecture at SanDisk. Daniel is a Product Management and Product Marketing professional with nine years of Product experience in the software industry. Prior to joining SanDisk, he was instrumental in product design with Interwise, establishing and managing the product management and the product marketing operations. Additionally, he has also managed technical sales into Microsoft, Avande and others along with driving technology partnerships with 3rd party vendors such as PeopleSoft and Docent.

This article and its contents copyright (c) 2005 by Daniel Shefer.

Additional articles by the author:

- [Pricing for Software Product Managers](#)
- [A Product Manager's Reading Anthology](#)
- [Ten Things Product Managers Need to Know About Sales](#)
- [Nine Things Product Managers Need to Know About Supporting Sales](#)
- [Disruptive Customer Demands](#)
- [Managing Desktop vs. Enterprise Applications](#)
- [OEMing Software](#)
- [Webcasts as a Lead Generation Tool](#)
- [Creating Effective Competitive Sales Tools for your Sales Reps](#)
- [Demos Pilots and the Sales Cycle](#)
- [Online Customer Forums](#)
- [The RFI as a Measurement of Product Marketing and Sales Reps Effectiveness](#)
- [Shortening the Sales Cycle](#)

This article and its contents copyright (c) 2005 by Daniel Shefer.