# SDL/DevSecOps



SDL，安全开发生命周期， 是微软提出并 于 2004 年正式应用的。

然后2006年，微软发布了基于SDL流程的第一个操作系统 Windows vista。

SDL 是微软基于其本身情况而产生的安全理念，符合自身，但是不一定符合其他公司。因为微软的产品就是操作系统，其具有长周期性、一次性产品、变更少等特点，所以产生了SDL的理念，不是DevSecOps理念。

SDL理念是由安全部门主导产品的生命周期，而不是由产品经理主导。安全在产品之初就在把控各节点的产品安全性，并且具有主导权，决定权。

只有这样才能真正落实SDL理念。安全的主导权是SDL的先天条件。

微软的安全部门获得这个条件的原因是，XP系统及之前系统的巨多的漏洞，对微软造成了极大极大的影响。

放眼望去，全中国，没有任何公司的安全部门有产品开发流程的主导权。

所以其他公司在用SDL理念进行安全介入的时候，就产生水土不服的感觉，也是为什么SDL不好落地的原因之一。

所以SDL其实不适合中国。

水土不服的原因是因为很多变化。

产品形态变化：传统系统，单机软件，到web系统到APP。

开发理念变化：瀑布模型，到敏捷模型，到devops模型。

开发技术栈变化：从混沌，到MVC，到微服务，到云原生。其技术栈本身也添加了很多安全理念。

这些变化导致，产品的安全修复成本，从最初的难以修复，变成了容易修复。修复难度在整个生命周期中是拉平的。

最初的开发技术栈导致，SDL是从安全培训开始，在前期，威胁建模在理念中占据了很大的篇章。威胁建模之后才能设计出来安全的产品架构，而且产品架构修改的成本极其巨大。但是现在的开发及运维技术栈，其本身就自带了高可用，可扩展，安全性等等特性。如MVC框架，如K8S，其理念是不断更新的。

所以安全理念在实际公司中的应用，应该是跟产品现状紧密相关的。

传统系统为主要产品，那么就用SDL理念。App为主要产品，就用devsecops理念。新能源汽车为主要产品，那么就取各种跟自己情况符合的理念来进行安全建设。

DevSecOps理念更适合中国的大环境。

DevSecOps理念中，只要求安全尽早参与，并没有要求安全的主导权。没有侵占产品经理的权力。

SDL理念中，没有包括已上线的持续长时间运营产品的安全工作。因为操作系统是一次性产品，产品本身没有长时间持续运营。产品一直有新版本，旧系统可以直接放弃维护。

DevSecOps理念中，就有持续的概念。对于无法重构的已上线的产品，安全工作可以有持续的安全监控，安全测试等手段来提高产品的安全性。

DevSecOps理念，适用性对于中国公司来说，更匹配一些。

有的行业SDL看起来也适合，比如游戏行业。但实际上还是有巨大差别的，一个是进程的主导权，一个是持续运营运维。

游戏的开发周期很长，跟系统类似，但是游戏的运营周期更长，系统可以直接放弃维护，但是游戏不能放弃维护。所以DevSecOps理念也更适合游戏。

所以安全理念在实际公司中的应用，应该是跟产品现状紧密相关的。

传统系统为主要产品，那么就用SDL理念。App为主要产品，就用devsecops理念。新能源汽车为主要产品，那么就取各种跟自己情况符合的理念来进行安全建设。