Atitit 最近十年来until2018软件开发领域的趋势 艾龙总结 attilax大盘点总结历史与趋势

[1. Keyword sec title 2](#_Toc22400)

[2. 语言本身： 2](#_Toc640)

[2.1. 工业标准 2 2](#_Toc17092)

[2.2. 2.第三方模块走红 3 2](#_Toc3936)

[2.3. 3.模块化编程和依赖管理 3 3](#_Toc19907)

[2.4. 4. 框架使用 3 3](#_Toc165)

[2.5. 5. 测试代码 4 3](#_Toc19918)

[2.6. 6.跨设备，跨平台 4 3](#_Toc25879)

[2.7. Dsl 声明式编程 函数式编程兴起 3](#_Toc21085)

[3. 工程方面的： 3](#_Toc8922)

[3.1. 1.工具化 4 3](#_Toc4967)

[3.2. 2. 工程化 5 3](#_Toc23263)

[3.3. 3. 自动化 5 更进一步的自动化。 4](#_Toc15245)

[3.4. Web已经开始工程化、模块化开发； 5](#_Toc11504)

[4. 其他方面： 5](#_Toc13426)

[4.1. 1. 版本控制，git，github 6 5](#_Toc2333)

[4.2. 2.生态圈意识 6 5](#_Toc3829)

[4.3. 1. WEB技术的桌面化和JavaScript的全栈化 7 5](#_Toc24981)

[4.4. 2. Web API的全面发展 7 5](#_Toc11820)

[4.5. 3. 语言之间的相互借鉴 8 5](#_Toc26667)

[4.6. 4. 语言解析器的工具化 8 5](#_Toc21334)

[4.7. 5. 数据交换语言的发展 8 5](#_Toc7443)

[5. 其他补充 5](#_Toc23056)

[5.1. 开源软件、开源社区的发展大大减少了程序员的重复劳动 5](#_Toc24746)

[5.2. 软件开发变得无比敏捷。瀑布开发模型已然是历史遗迹 5](#_Toc12187)

[5.3. 大数据 数据挖掘 6](#_Toc2107)

[5.4. 云计算的普及。10年间，云服务从一个云里雾里的概念发展成为软件产品交付的主要模式。 6](#_Toc9464)

[5.5. 分布式 程序运行范式的变化。 从中心化走向分散化 6](#_Toc13850)

[5.6. DSL ui h5 db sql 6](#_Toc31513)

[5.7. 浏览器是一个平台，重要性不亚于操作系统本身。 6](#_Toc30619)

[5.8. 前端，还有了全栈 6](#_Toc16772)

[5.9. 十年前只有网站和网页，十年后一切都成了Application， 6](#_Toc32680)

[5.10. 单元测试 7](#_Toc8981)

[5.11. Web2.0 ugc在线知识库 7](#_Toc19192)

[5.12. 云 vps 7](#_Toc22975)

[5.13. 云与服务化 DBaaS服务 xaas 7](#_Toc9030)

[5.14. Msa架构微服务架构 7](#_Toc2438)

[5.15. mobile 开发的流行 7](#_Toc24824)

[5.16. 版本迭代加快 8](#_Toc1024)

[5.17. dogfood 等概念的成熟，大家对于测试的界限越来越模糊。 8](#_Toc1793)

[5.18. 十年前 Github 和 Stack Overflow 都还不存在。 8](#_Toc16669)

[5.19. Web 取代 cs 8](#_Toc29470)

[5.20. 关系型数据库已经衰落了。 mongodb崛起 redis 8](#_Toc32527)

[5.21. Js崛起 vm编译型语言开始衰落 8](#_Toc27791)

[5.22. 并发 多核处理器的兴起，编程语言需要更好的利用并发以提高性能 8](#_Toc19075)

[5.23. 基础理论没有重大发展，工程化，模块化发展迅速。端与端的分离，各种框架 8](#_Toc28155)

[5.24. 引擎的横空出世，让游戏开发也变得轻巧许多 8](#_Toc31113)

[5.25. Ui dsl分离 h5 8](#_Toc11048)

[5.26. other 9](#_Toc14050)

# Keyword sec title

十年来软件行业的大进步

从这个角度来看，今天的测试驱动开发，它提升效率的根本就是把测试用例变得

# 语言本身：

## 工业标准 2

网页标准有w3c控制，尤其是浏览器的开发，所有主流的浏览器都会自觉符合这个组织的标准，当然这些开发商本身就是这个组织的成员。所以新的HTML5，CSS3，ES6 javascript的新特性的得到顺利推动，让大部分主流浏览器都支持它，w3c功不可没.  
  
PHP有PHPFIG组织，虽然不是强制性的，但是很多新的框架和库都自觉遵守这个组织的编程标准

## 2.第三方模块走红 3

各种语言的框架和库，可能比自己的语言还出名，比如css的Bootstrap，javascript的jQuery；一个好的框架和库甚至可以推动这个这个语言的发展，比如说PHP的Laravel框架，Javascript的jQuery.

## 3.模块化编程和依赖管理 3

## 4. 框架使用 3

## 5. 测试代码 4

## 6.跨设备，跨平台 4

## Dsl 声明式编程 函数式编程兴起

# 工程方面的：

## 1.工具化 4

Ruby社区贡献的自动化工具链，大量地被移植到其他语言平台上

我觉得工具化非常突出了，凡是能工具完成的事情，绝对不手工完成。以下几个方面都是可以找到相应工具，帮助开发者管理代码质量

代码风格检查

工业标准检查

代码整理

代码复杂度检查

单元测试覆盖率检查

依赖管理

压缩代码

重复代码检查

无用代码检查

## 2. 工程化 5

工程化是以工具化为基础的，没有工具，那么工程化也无从谈起。  
  
工程的核心就是流程自动化，又称之为构建，这些包括了：代码质量检测，代码压缩，代码合并，代码优化，代码编译，单元测试等等部分。构建就是把这些以工作流程的方式组合起来，然后用一个命令行运行这整个流程。它有点像批处理，但是是程序开发中使用的特殊批处理

## 3. 自动化 5 更进一步的自动化。

自动化是以工程化为基础的，工程化本身就是一种流程自动化。而自动化有在工程化的过程中更进一步的自动化。  
  
持续集成就是全自动化的一个终极体现。他的主要流程是：版本控制库->构建->测试->报告.  
持续集成有点像windows的定时任务，但是它是程序开发专用的定时任务。

另外还有个持续部署，其实就是持续集成在测试成功后部署上产品服务器上的流程。如今有些网站一天就要部署几十次，有了持续部署后，部署多少次都毫无压力。  
  
工具化，工程化，自动化的关系挺有意思，前者是后者的基础，而后者却极大推动了前者的发展。它们是相互积极作用，相互推动了对方的发展，形成了一个很好的良性循环

从这个角度来看，今天的测试驱动开发，它提升效率的根本就是把测试用例变得可复用和可扩展。当代码变得更复杂时，测试用例可以很好地让开发速度稳定持续下去，而不是担心各种潜在的bug。  
至于app store 的dogfood，服务器的快速部署，这些都是这些事情自动化以后的表面现象。

Dw就是可视化拖拽式h5 ide呀。

## Web已经开始工程化、模块化开发；

# 其他方面：

## 1. 版本控制，git，github 6

## 2.生态圈意识 6

## 1. WEB技术的桌面化和JavaScript的全栈化 7

## 2. Web API的全面发展 7

## 3. 语言之间的相互借鉴 8

## 4. 语言解析器的工具化 8

## 5. 数据交换语言的发展 8

# 其他补充

## 开源软件、开源社区的发展大大减少了程序员的重复劳动

## 软件开发变得无比敏捷。瀑布开发模型已然是历史遗迹

遗迹，持续集成、持续交付、微服务、DevOps这些工程实践，让软件的发布速率提高了一个甚至数个数量级

## 大数据 数据挖掘

数据成为新的宝藏。互联网的发展和智能手机的普及让人类累积了大量的数据，而这些数据的挖掘和探索，将会产生无法想象的价值

## 云计算的普及。10年间，云服务从一个云里雾里的概念发展成为软件产品交付的主要模式。

AWS，Heroku, Saleforce，Slack这些公司的成功让IaaS/PaaS/SaaS/CaaS(容器云)的价值都得到了充分验证。对于互联网企业来说，开源降低了研发成本，而云服务进一步降低了运营成本，初创互联网公司如雨后春笋，大众创业成为可能。

## 分布式 程序运行范式的变化。 从中心化走向分散化

虽然现在大部分互联网产品仍然遵循客户端－服务器的范式。但是分布式的软件产品已经初露端倪，区块链、比特币的出现不单是金融领域的创新，也可能预示着软件产品从中心化走向分散化的一种趋势

## DSL ui h5 db sql

## 浏览器是一个平台，重要性不亚于操作系统本身。

现在的普通JS程序动辄几万行，可能有着所有界面应用开发里最好用的调试工具：Chrome浏览器和它的developer tools。十年前主流的浏览器是IE6，只是一个浏览器，今天的浏览器是一个平台，重要性不亚于操作系统本身。

## 前端，还有了全栈

## 十年前只有网站和网页，十年后一切都成了Application，

而application概念在十年前专指你从windows开始菜单能打开的那些。

## 单元测试

十年前测试主要是人工的集成，单元测试在大多数开发团队里还只是道听途说的概念，更不要说Continuous Integration。经过十年，单元测试的实践已经深入人心。测试本身已经是一个单独的产业，各种工具，服务被创造出来，现在的软件，从操作系统到mobile app的质量，比十年前的产品已经不可同日而语。

## Web2.0 ugc在线知识库

十年前博客还方兴未艾，Youtube还只是一个玩具。十年里博客已经是数字生活的常态，WordPress的安装覆盖全世界，Youtube不再是一个网站，而是地球上史无前例的多媒体平台。十年前学编程主要看书，十年后的今天，学任何技术都能找到优质的在线免费内容，自我教育只受限于意愿而不是金钱

## 云 vps

十年前中小企业架设网站的唯一选择是虚拟主机，当时新网一个很烂的虚拟主机一年卖几千块。十年后的今天，绝大部分的中小企业业务已经迁移进某个云里。懂点技术的人可以在十分钟内完成主机购买到网站上线的操作。

## 云与服务化 DBaaS服务 xaas

十年前面对数据库编程就是mysql，十年后互联网上你能找到一堆成熟的的DBaaS服务。

## Msa架构微服务架构

十年前教科书上还在大讲特讲三层结构，十年后的今天，一切都是服务，服务，服务。面向服务而架构，面向服务开发，面向服务来部署，硬件是服务，软件也是服务。

## mobile 开发的流行

## 版本迭代加快

## dogfood 等概念的成熟，大家对于测试的界限越来越模糊。

## 十年前 Github 和 Stack Overflow 都还不存在。

## Web 取代 cs

## 关系型数据库已经衰落了。 mongodb崛起 redis

## Js崛起 vm编译型语言开始衰落

## 并发 多核处理器的兴起，编程语言需要更好的利用并发以提高性能

## 基础理论没有重大发展，工程化，模块化发展迅速。端与端的分离，各种框架

## 引擎的横空出世，让游戏开发也变得轻巧许多

## Ui dsl分离 h5

用户界面不再是写死在代码里，而是将界面描述，与呈现进行了分离。

比如网页的描述语言就是HTML，浏览器作为HTML的呈现引擎。

桌面程序界面的描述语言就是XAML（WPF编程技术）

报表的描述语言就是RDL。

## other

当年以为计算机专业的，没有其他的行业知识会很难找工作，现在发现任何行业都招计算机专业的人。

7.当年以为，软件这东西像写小说，写着写着就没什么可写了。现在发现，软件这东西，像拍电影，拍了再多次还可以拍续

十年前国内一线城市应届程序员起薪2-3k遍地都是，十年后，北上广一线毕业生码农起薪5-6k起，不过我可能还是太保守，据说8-10k了。