一个完整的**高效的产品团队**的工作模式主要分为以下几个方面：

产品研发角色

一个比较完整的互联网产品会由以下人员构成：

PM：产品经理

UI：视觉设计师

UE：交互设计师

UR：用户研究员

FE：前端工程师

BE：后端工程师

OP：运维工程师

**那么他们的职责有哪些?**

PM：完成产品的构思，决策，规划，输出含有项目功能点的产品需求文档

UI：负责产品的视觉风格设计

因为设计师对视觉、颜色的敏感度要比前后端、测试强很多，有时候测试也发现不了某个色值在网页和设计稿的具体差异，所以需要进行设计效果确认；

UE：负责产品的交互设计

UR：负责设计和实施各种客户满意度、需求等调研项目，分析相关数据，输出强有力研究报告，协助产品经理改进产品

FE：负责高保真还原UI的设计搞以及UE的交互，配合BE完成接口数据的联调和上线

BE：编写后台程序，设计数据库，和FE一起完成实现PM的产品功能点

OP：负责线上和测试机器的环境搭建和维护

创业公司人员往往身兼数职，比如这些角色常常会由一个同学担任UI/UE，PM/UR，FE/BE，BE/OP，当然随着公司的发展，这些角色会逐渐单一化和专业化，最后更会以独立团队出现。

**产品研发流程**

业务驱动的公司一般是由业务方来提出业务需求，比如marketing 人员；

而产品驱动的公司一般由产品经理直接来提出需求，具体的研发流程如下：

1.产品经理进行功能需求的分拆和细化，输出带有产品功能点的产品原型。用研会对相关产品原型的需求进行用户调研和问卷调查，协助产品对需求做好合理有效的把控；

3.交互和视觉根据产品原型的功能点进行视觉设计和交互设计，输出视觉稿和交互原型；

4.前端工程师 根据视觉设计稿和交互原型，输出带有完整的交互的**高保真静态页面**；

5.后端进行开发，包括需求分析、设计、编码等环节；

6.前后端进行联调，联调的差不多时候，会把项目测试地址发出来review：

产品进行功能点把控，遗漏的功能点进行补充开发；

视觉设计师，因为设计师对风格和颜色的敏感度要强很多，所以需要他们进行设计效果确认；

然后提测。

7.测试工程师在项目设计阶段编写测试用例，收到前后端的测试代码，进行功能测试、UI测试、接口测试、安全测试以及兼容性测试；

8.上线阶段分为两种：

第一种，前后已经分离的，前端先独立上线新版号的静态资源至CDN，后端再次上线，最后修改静态资源的线上引用版本号；

第二种，前后工程化未分离的，静态资源放在后端代码工程里，那么前后端会一起上线；

**前端团队角色管理：**

1.归属：聚合和分散

>>聚合：前端人员工位集中在一起，需求由前端经理统一分配，团队归属于UED，当然也会细化出几个业务小组，对接不同的业务线，方便管理。聚合的好处是管理非常方便，由于前端人员在公司里相对较少，聚合也会使前端团队成员的归属感非常强，同时相关前端问题可以集中高效解决

>>分散：相对聚合来讲，分散后前端团队一般归属于大业务线的研发经理统一管理，可能是由原来UED前端团队拆分到各个业务线里，也可能是大部门自己组建成独立前端团队，有一个问题就是分拆后团队间沟通，成长不如聚合流畅

2.职级：manager/ group leader/ Senior Enginer/ Junior Enginer

3.职责：页面制作/脚本开发

>>页面制作：高保真还原视觉设计和页面效果

>>脚本开发：负责页面交互的技术实现，配合后端完成数据接口的前台调用

4.周边：H5/动效开发/小游戏开发/iOS、Android

**前端开发流程**

项目构建：生成标准化项目文件目录

模块引入：引入已开发好并且可以在当前项目中使用的模块

模块编译：把模块依赖的模板文件/js/css/json数据文件引入页面中，scss编译成css，数据打入模板生成html

输出：主要是进行性能优化，静态资源加上cdn，css sprite，js/css/图片压缩

联调：把前端本地的静态文件上传至测试服务器或者后端机器

上线：把项目中的静态资源包括js/css/图片上线至CDN

**前端工具系统**

之开发环境：

>>分为本地环境、测试环境、线上环境

前端协作模式/流程如下：

需求评审后

前后端定数据接口(前后分离)

前端页面制作/交互实现

后端开发

前后端联调

前端 / 后端上线

**前后端分离**可以这样处理：

前端可在本地基于json数据文件写模板，后端也可以基于json数据文件写后端业务逻辑相关的数据接口

**前端工程化**

百度百科关于“软件工程”名词解释：软件工程是一门研究用工程化方法构建和维护有效的、 实用的和高质量的软件的学科。

**前端工程化要解决哪些问题：**

适合自身团队发展的开发流程和开发规范，包括代码规范、模块化规范（提高生产力）

一套自动化的项目发布/部署方案（提高系统灵活性）

一套自动化代码质量检测方案（提高系统可靠性）：开发阶段的Eslint，summit的交叉review，技术review，架构review等

比较好的性能优化（提升用户体验）

**工程化解决方案要点**

前端命令行工具：工具编译引擎、核心模块、扩展等；

前端模块：模块的定义、安装、创建、调试、预览、发布，前端模块构建平台建设；

前端组件：基础库和模块加载器的选型、组件库的建立；

前端规范：编码规范、目录规范、组件规范等；

前端文档：前端必备的文档；

前端开发流程：项目生成、编译、输出、联调、部署、上线；

前端性能优化：pc和h5页面性能优化的方式和思路；

要能够提供：

Html及模板数据编译，css(sass、less)的compile，TS-ES6/7的compile，兼容SVG/sprite/font-awesome ，自动压缩优化js、css、png文件（base64 ） 等

支持本地静态文件预览，内置本地开发调试服务器，以及当前目录浏览

支持实时监听文件，文件被修改时会自动编译成css，并刷新浏览器

实时在控制台输出错误信息，方便定位代码错误

生成标准化的项目文件夹并统一 文件编码，即无论当前文件格式是gbk，gb2312，utf8，utf8-bom，统一输出utf8

支持本地，联调，测试及线上环境，支持版本控制及CDN内容分发控制，例如在本地开发需支持：

支持html/js/css文件格式化，支持html/js/css代码压缩，支持html/js/css文件lint，代码质量检查

每个项目都统一配置，统一调用公共框架

模块开发方面， 可快速方便的对模块进行创建，引用，预览，安装和发布。通过积累，可形成完全符合自己业务的模块云服务

以上就是一个完整高效的前端工作团队和协作模式。