* Phân mảnh là miêu tả ứng dung không có khả năng giúp viết 1 lần và có thể chạy ở bất cứ nơi nào.
* Nguyên nhân dẫn đến hiện tượng này bao gồm sự đa dạng và khác nhau ở một số điểm của phần cứng; sự đa dạng của các phần mềm sử dụng (các nền tảng khác nhau và các tiêu chuẩn thực hiện khác nhau); các tính năng được viết lại, mod lại tạo ra các biến thể khác nhau tối ưu cho từng đối tượng sử dụng khác nhau; sở thích và kiến thức cũng như thôi quen của người dung; cuối cùng là sự đa dạng của môi trường triển khai ảnh hưởng từ những nguyên nhân xung quanh như thông số kĩ thuật chung của khu vực hoặc yêu cầu môi trường chuẩn của khu vực hoặc công việc.
* Các quy tắc cần phải thực hiện khi thực hiện và triển khai một phần mềm di động như trong quy trình hợp nhất của IBM là: mô hình hoá kinh doanh=> quản lý yêu cầu => phân tích và thiết kế => triển khai => kiểm tra => quản lý dự án => quản lý cấu hình và thay đổi => môi trường. Những thao tác này làm tiêu tốn thêm nhiều thời gian và tài nguyên và có thể gây ra phát sinh lỗi cũng như lỗ hổng để khai thác. Đó là các nguyên nhân chính khiến ta phải chú ý trong quá trình phát triển phần mềm di động.
* Hiện nay thì các ứng dụng di động thường có nhiều phiên bản để phù hợp với các thiết bị khác nhau do sự đa dạng thiết yếu của các thiết bị ( kích thước, độ phân giải màn hình, …) và các sự đa dạng tình cờ ( các lỗi khi triển khai API). Do đó cần phải tiếp cận và nguyên cứu cũng như thực hiện các bước chuẩn hoá tốt để có thể giảm thiểu được các sự đa dạng tình cờ giúp hạn chế hao tổn tài nguyên một cách hiệu quả nhất và ít phát sinh lỗ hổng gây lỗi hoặc bị khai thác bởi bên thứ 3.
* Các các tiếp cận phân mảnh:
  + MANUAL-MULTI: nói đơn giản có thể hình dung là đối với mỗi OC thì ta thiết kế 1 phiên bản riêng biệt. Nhưng đây là một phương pháp tốn nhiều tài nguyên và thời gian nhưng lại có nhiều công việc bị trùng lặp.
  + DERIVE-MULTI: đây là phương pháp xây dựng phiên bản dành cho từng OC theo cơ sở mã có sẵn và được triển khai theo 3 cách: đóng gói chọn lọc, lập trình meta và tạo tự động. Đây là phương pháp tối ưu hơn hạn chế được thời gian và tài nguyên vào những thứ lặp đi lặp lại không cần thiết.
  + SINGLE-ADAPT: đây là phương pháp xây dựng một phiên bản tối ưu được cho tất cả các OC và nó được chia thành 2 phần đó là FITS-ALL và ALL-IN-ONE.
* Các khía cạnh khác của vấn đề phân mảnh bao gồm phân mảnh trong các ứng dụng truyền tống, phân mảnh trong các dòng sản phẩm, phân mảnh trong ứng dụng sms và web di động, các ưng dụng java và không phải java.
* Trong tương lai thì các vấn đề liên quan tới phân mảnh cũng được tang cao như khả năng của các thiết bị được nâng cao, các ứng dụng di động yêu cầu có khả năng chạy trên máy tính để bàn, số lượng mẫu thiết bị di động tang mạnh, và sự chuẩn hoá trong quá trình phát triển.
* Phân mảnh là một vấn đề quan trọng trong quá trình phát triển một ứng dụng di động, tuỳ thuộc và khả năng cũng như quy mô dự án mà ta có thể áp dụng các phương thức tiếp cận cũng khư khắc phục khác nhau nhưng nhìn chung thì đây là vấn đề mà bất cứ ứng dụng nào cũng phải đối mặt. Đối với các phương thức trên tôi thấy mỗi cái đều có ưu điểm và nhược điểm riêng, do đó tuỳ vào tình huống và quy mô dự án mà ta có thể áp dụng một cách linh hoạt để có thể có được kết quả tốt nhất mà ít tốn tài nguyên và thời gian nhất.